

---

---

**Information technology — Security  
techniques — Digital signatures with  
appendix —**

**Part 2:**  
**Identity-based mechanisms**

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

*Partie 2: Mécanismes basés sur des identités*



## 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 14888-2 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*

Further parts may follow.

Annexes A and B of this part of ISO/IEC 14888 are for information only.

# Information technology — Security techniques — Digital signatures with appendix —

## Part 2: Identity-based mechanisms

### 1 Scope

ISO/IEC 14888 specifies a variety of digital signature mechanisms with appendix for messages of arbitrary length and is applicable in schemes providing entity authentication, data origin authentication, non-repudiation, and integrity of data.

This part of ISO/IEC 14888 specifies the general structure and the fundamental procedures which constitute the signature and verification processes of an identity-based digital signature mechanism with appendix for messages of arbitrary length.

### 2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14888. At the time of publication, the editions indicated were valid. All standards are subject to revision, and 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 standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 9796: 1991, *Information technology - Security techniques - Digital signature scheme giving message recovery*.

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

### 3 General

The verification of a digital signature requires the signing entity's verification key. It is therefore 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 digital signature scheme is said to be "identity-based."

The key generation process of the identity-based mechanism defined in this part of ISO/IEC 14888

involves a trusted third party. The trusted third party has a secret parameter, the key generation exponent, which it uses to derive the signature keys of other entities. The secrecy of the signature keys depends unconditionally on the secrecy of the key generation exponent.

In the verification of an identity-based signature, two parameters are needed. The first one, the domain verification exponent, is common to all entities, while the second one, the signing entity's verification key, is specific to each entity. In the identity-based mechanism defined in this part of ISO/IEC 14888, an entity's verification key is derived directly from the entity's identification data, using a public function.

The identity-based signature mechanism with appendix is an example of a randomized mechanism, as described in ISO/IEC 14888-1. The descriptions of the signature and verification processes follow the general procedures specified in Clause 10 of ISO/IEC 14888-1. In particular, this part makes use of the general requirements, definitions and symbols given in ISO/IEC 14888-1.

The identity-based digital signature mechanism with appendix is defined by the specification of the following processes:

- key generation process;
- signature process; and
- verification process.

### 4 Definitions

For the purposes of this part of ISO/IEC 14888, the following definitions apply.

#### 4.1

##### domain modulus

a domain parameter, which is a positive integer resulting from the product of two distinct primes which are known only to the trusted third party

#### 4.2

##### domain verification exponent

a domain parameter which is a positive integer

**4.3****key generation exponent**

a positive integer known only to the trusted third party

**4.4****public key derivation function**

a domain parameter, whose function is to map strings of bits into positive integers

NOTE 1 This function is used to transform an entity's identification data into the entity's verification key, and satisfies the following two properties.

- It is computationally infeasible to find any two distinct inputs which map to the same output.
- Either the probability that a randomly chosen value  $Y$  is in the range of the function is negligibly small, or for a given output it is computationally infeasible to find for a given output an input which maps to this output.

NOTE 2 Negligibility and computational infeasibility depend on the specific security requirements and environment.

**4.5****trusted third party**

A security authority, or its agent, trusted by other entities with respect to security related activities

**5 Notation****5.1 Symbols**

In addition to the symbols defined in ISO/IEC 14888-1, the following symbols are used.

|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| $D$                | Key generation exponent                                                  |
| $I$                | Identification data                                                      |
| $N$                | Domain modulus                                                           |
| $P, Q$             | Prime numbers                                                            |
| $V$                | Domain verification exponent                                             |
| $y$                | Public key derivation function                                           |
| $\text{lcm}(A, B)$ | The least positive integer such that $C \bmod A = 0$ and $C \bmod B = 0$ |
| $C$                | $\equiv 0 \pmod{B}$                                                      |

**6 Key production process**

The key production process of this identity-based signature mechanism is an application of the signature scheme specified in Annex A of ISO/IEC 9796-1. It consists of the following three procedures:

- generating domain parameters; and
- generating signature key.

An entity shall be chosen to act as a trusted third party. Using its own secrets, the trusted third party

generates the private signature key for each entity as a function of that entity's identification data.

In some instances the validation of domain parameters and keys, may be required. However, it is outside the scope of this part of ISO/IEC 14888.

**6.1 Generating domain parameters**

This procedure is executed once when the domain is set up.

The trusted third party produces the domain verification exponent  $V$  and the domain modulus  $N$ . The domain verification exponent shall be chosen to be an odd integer. The domain modulus shall be the product of two large prime integers  $P$  and  $Q$  such that  $P - 1$  and  $Q - 1$  are coprime to  $V$ . Furthermore, the trusted third party determines the key generation exponent, which is the least positive integer  $D$  such that  $DV - 1$  is a multiple of  $\text{lcm}(P - 1, Q - 1)$ .

Specifically,  $D$  shall be such that:

$$U^{DV} \bmod N = U$$

for all integers  $U$ ,  $0 < U < N$ .

The public domain parameters are  $N$  and  $V$ . The trusted third party keeps  $D$  for its own use. It shall be computationally infeasible for other entities to compute  $D$  from  $N$  and  $V$ .

NOTE — The values of  $N$  and  $V$  shall be chosen large enough to satisfy the specific domain security requirements. The length of  $N$  varies typically between 1024 bits and 2048 bits while the length of  $V$  is recommended to be at least 80 bits.

Let  $T$ ,  $T'$  and  $L$  be integers such that  $T - T' = LV$ . Let  $X$  be a signature key and  $Y$  be the corresponding verification key as defined in 6.2. Then  $X^{T'} \equiv X^T \cdot Y^L \bmod N$ .

Given a signature  $(R, S)$  corresponding to assignment  $T$ , one can then produce the signature  $(R, S')$  corresponding to assignment  $T'$  by computing  $S' = S \cdot Y^L \bmod N$  (See 7.4). Therefore,  $V$  must be chosen sufficiently large so that given  $T$ , the probability that a randomly chosen  $T'$  satisfies  $T - T' \equiv 0 \bmod V$  is sufficiently small.

The set of the domain parameters of an identity-based signature mechanism with appendix also includes a public key derivation function  $y$ , which is used to transform a signing entity's identification data into a positive integer less than  $N$ .

**6.2 Producing the verification key and generating the signature key**

Each user entity has unique identification data. To generate a private signature key for an entity with identification data  $I$ , the trusted third party first

computes the verification key  $Y$  from the public key derivation function  $y$  and the identification data  $I$ .

$$Y = y(I).$$

NOTE — The probability that  $Y$  is equal to zero or an integer multiple of  $P$  or  $Q$  is negligible, provided that  $P$  and  $Q$  are sufficiently large.

The trusted authority computes the private signature key  $X$  as

$$X \equiv Y^{-D} \bmod N.$$

As a result of the key generation process, the entity with identification data  $I$  has a signature key  $X$  which satisfies the equation,

$$X^V \cdot y(I) \bmod N \equiv 1.$$

In an identity-based signature mechanism, the verifier obtains the knowledge of the signing entity's verification key from the signing entity's identification data by computing  $Y = y(I)$ . Once the verification has been computed, it can be cached for further use.

## 7 Signature process

In this clause, the signature process of an identity-based signature mechanism is described. The mechanism is randomized and follows the general model for randomized signature mechanisms described in ISO/IEC 14888-1. The signature process consists of the following procedures:

- producing pre-signature;
- preparing message;
- computing witness; and
- computing signature.

In this process the signer makes use of its signature key  $X$  and the public domain parameters  $N$  and  $V$ .

The result of this process is the signature  $\Sigma$  which has two parts,  $R$  and  $S$ . The signing entity optionally forms a text field, containing the entity's identification data. The signature and the optional text field form the appendix, which is appended to the message by the signer.

### 7.1 Producing pre-signature

The pre-signature procedure of an identity-based signature mechanism consists of two steps:

- producing randomizer  $K$ ; and
- computing pre-signature  $\Pi$ .

#### 7.1.1 Producing randomizer

The signing entity generates a randomizer, which is an integer,  $K$  with  $0 < K < N$ . Depending on the mechanism, there may be additional constraints on

this generation procedure. The output of this step is  $K$ , which the signing entity keeps secret.

#### 7.1.2 Computing pre-signature

The input to this step is the randomizer  $K$ . The output is the pre-signature  $\Pi$  computed as:

$$\Pi = K^V \bmod N.$$

### 7.2 Preparing message

Two data fields,  $M_1$  and  $M_2$  are derived from the message  $M$  as described in 8.2 of ISO/IEC 14888-1.

The process of preparing the message shall satisfy one of the following two conditions:

- the entire message shall be re-constructible given  $M_1$  and  $M_2$ ; or
- it shall be computationally infeasible to find two distinct messages  $M$  and  $M'$  such that the derived pairs  $(M_1, M_2)$  and  $(M'_1, M'_2)$  are equal.

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

### 7.3 Computing witness

The deterministic witness is computed as a hash-token  $H$  of  $M_1$  using a collision-resistant hash-function (see Figure 1).

The randomized witness  $R$  is defined as the concatenation of an optional field, which can be used to identify the hash-function and padding method and is computed using a collision-resistant hash-function.  $R$  depends on the pre-signature  $\Pi$  and optionally, on  $M_1$ , if  $M_1$  is not empty (see figure 2).

NOTE — The hash-function identifier shall be included in the hash-token unless the hash-function is uniquely determined by the signature mechanism or by the domain parameters.

### 7.4 Computing signature

The computation of a signature in an identity-based signature mechanism consists of the following steps:

- computing the first part of the signature;
- computing assignment; and
- computing the second part of the signature.

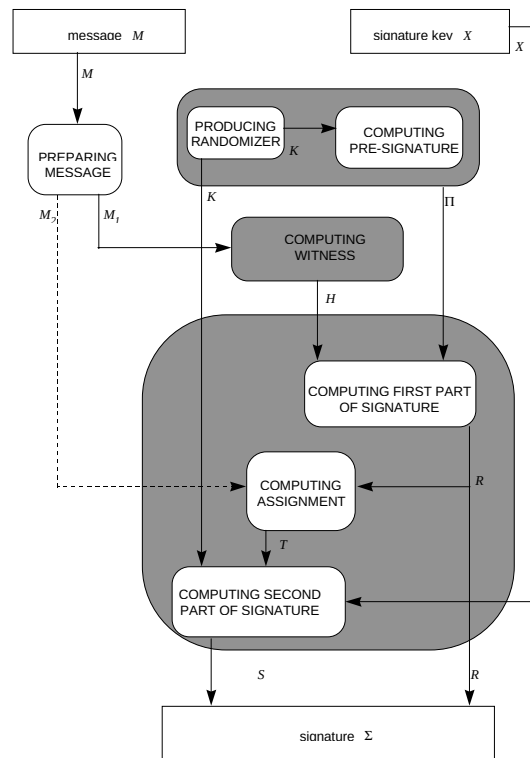


Figure 1 — Signature process with a deterministic witness

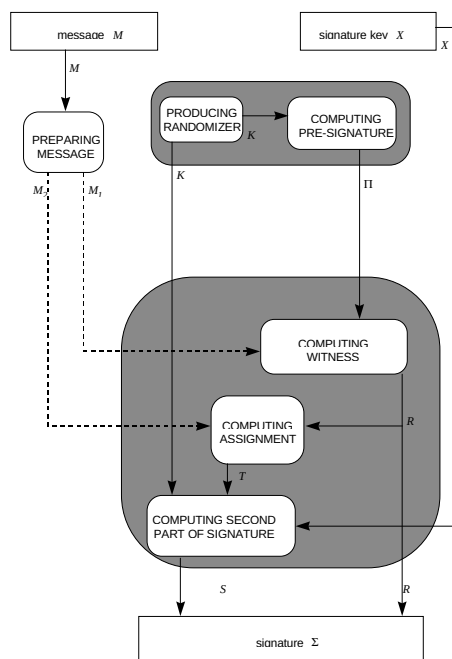


Figure 2 — Signature process with a randomized witness

#### 7.4.1 Computing the first part of the signature

If the witness is deterministic, the first part  $R$  of the signature is computed as a function of  $H$  and  $\Pi$ . This function is invertible in the following way.

- Given  $\Pi$  and  $R$ , the hash-token  $H$  can be recovered (see Figure 1).

If the witness is randomized, it is the first part of the signature  $R$  and no further computation is needed (see Figure 2).

NOTE — The domain parameters must include an agreed upon method for converting a string of bits to a positive integer, in order that this step can be carried out.

#### 7.4.2 Computing assignment

The assignment  $T$  is a positive integer and computed as a function of the first part of the signature. The assignment also depends on the second part  $M_2$  of the message, if  $M_2$  is not empty.

It is required that the method of computing the pair  $(R, T)$  satisfies the following condition.

It is computationally infeasible to find any two pairs  $(M, \Pi)$  and  $(M', \Pi')$  with the same resulting pair  $(R, T)$ .

#### 7.4.3 Computing the second part of the signature

The signature function of the identity-based signature mechanism has the following form,

$$S = K \cdot X^T \bmod N,$$

where  $K$  is the randomizer computed in 7.1.1,  $T$  is the assignment computed at 7.4.2,  $X$  is the signature key, and  $N$  is the domain modulus. The output of the signature function is the second part  $S$  of the signature.

## 8 Verification process

The verification process consists of the following procedures:

- preparing message;
- retrieving witness;
- computing verification function; and
- verifying witness.

At the beginning of the verification process the verifier must have the values of the following data items available:

- domain parameters  $N$  and  $V$ ;
- the signer's verification key  $Y$ ;
- message  $M$ ;
- signature  $\Sigma = (R, S)$ ; and
- optional text (from appendix).

A successful verification of a signature implies that the signature was created using a signature key  $X$  which corresponds to the verification key  $Y$ .