

INTERNATIONAL  
STANDARD

ISO/IEC  
9796

First edition  
1991-09-15

**Information technology — Security  
techniques — Digital signature scheme giving  
message recovery**

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



Reference number  
ISO/IEC 9796:1991(E)

## Contents

	Page
Foreword .....	iii
Introduction .....	iv
<b>1 Scope .....</b>	<b>1</b>
<b>2 Definitions .....</b>	<b>1</b>
<b>3 Symbols and abbreviations .....</b>	<b>1</b>
<b>4 General overview .....</b>	<b>1</b>
<b>5 Signature process .....</b>	<b>2</b>
<b>6 Verification process .....</b>	<b>3</b>

## Annexes

<b>A Example of a public-key system for digital signature .....</b>	<b>4</b>
<b>B Illustrative examples related to annex A .....</b>	<b>6</b>
<b>C Some precautions taken against various potential attacks related to annex A .....</b>	<b>11</b>
<b>D Bibliography .....</b>	<b>12</b>

© ISO/IEC 1991

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland  
Printed in France

## 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 9796 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Annexes A, B, C and D are for information only.

## Introduction

A digital signature in electronic exchange of information is a counterpart to a handwritten signature in classical mail.

Most digital signature schemes are based upon a particular public-key system. Any public-key system includes three basic operations:

- a process producing pairs of keys: a secret key and a public key;
- a process using a secret key;
- a process using a public key.

In any public-key digital signature scheme, the secret key is involved in a signature process for signing messages, and the public key is involved in a verification process for verifying signatures. A pair of keys for a digital signature scheme thus consists of a "secret signature key" and a "public verification key".

Two types of digital signature schemes are clearly identified.

- When the verification process needs the message as part of the input, the scheme is named a "signature scheme with appendix". The use of a hash-function is involved in the calculation of the appendix.
- When the verification process reveals the message together with its specific redundancy (sometimes called the "shadow of a message"), the scheme is named a "signature scheme giving message recovery".

This International Standard specifies a scheme for digital signature of messages of limited length.

This digital signature scheme allows a minimal resource requirement for verification. It does not involve the use of a hash-function and it avoids the known attacks against the generic algorithm in use.

The message need not be in a natural language. It may be any arbitrary string of bits of limited length. Examples of such messages are cryptographic key materials and the result of hashing another, longer message, which is also called the "imprint of a message". A characteristic example is a structured set of a few strings of bits generated by cryptographic software and hardware, one of these strings coding control information produced within the hardware.

NOTE — The use of this International Standard may involve patented items.

# Information technology — Security techniques — Digital signature scheme giving message recovery

## 1 Scope

This International Standard specifies a digital signature scheme giving message recovery for messages of limited length and using a public-key system.

This digital signature scheme includes

- a signature process using a secret signature key and a signature function for signing messages;
- a verification process using a public verification key and a verification function for checking signatures while recovering messages.

During the signature process, messages to be signed are padded and extended if necessary. Artificial redundancy is then added, depending upon the message itself. No assumption is made as to the possible presence of natural redundancy in the messages. The artificial redundancy is revealed by the verification process. The removal of this artificial redundancy gives message recovery.

This International Standard does not specify the key production process, the signature function and the verification function. Annex A gives an example of a public-key system including key production, signature function and verification function. The various steps of these operations are illustrated by examples in annex B.

Some parameters in the scheme are related to security: this International Standard does not specify the values to be used in order to reach a given level of security. However, this International Standard is specified in such a way as to minimize the required changes in its use if some of these parameters have to be modified.

## 2 Definitions

For the purposes of this International Standard, the following definitions apply.

**2.1 message:** String of bits of limited length.

**2.2 signature:** String of bits resulting from the signature process.

## 3 Symbols and abbreviations

<i>MP</i>	Padded message
<i>ME</i>	Extended message
<i>MR</i>	Extended message with redundancy
<i>IR</i>	Intermediate integer
$\Sigma$	Signature
<i>k<sub>s</sub></i>	Length of the signature in bits
<i>IR'</i>	Recovered intermediate integer
<i>MR'</i>	Recovered message with redundancy
<i>MP'</i>	Recovered padded message
<i>Sign</i>	Signature function under control of the secret signature key
<i>Verif</i>	Verification function under control of the public verification key
<i>mod z</i>	Arithmetic computation modulo z
$\mu$	Nibble
$\Pi$	Permutation of the nibbles
<i>m</i>	Byte
<i>S</i>	Shadow of the bytes
<i>XII Y</i>	Concatenation of strings of bits <i>X</i> and <i>Y</i>
<i>X<math>\oplus</math> Y</i>	Exclusive-or of strings of bits <i>X</i> and <i>Y</i>

### NOTES

1 All integers (and all strings of bits or bytes) are written with the most significant digit (or bit or byte) in left position.

2 The hexadecimal notation, with the digits 0 to 9 and A to F, is used in table 1 and in annex B.

## 4 General overview

The next two clauses specify

- the signature process in clause 5;
- the verification process in clause 6.

Each signing entity shall use and keep secret its own signature key corresponding to its own public verification key.

Messages to be signed shall be padded and extended if necessary. Redundancy is then added according to rules specified in clause 5. From the extended messages with redundancy, signatures shall be computed using the secret signature key as specified in clause 5.

Each verifying entity should know and use the public verification key specific to the signing entity. A signature shall be accepted if and only if the verification process specified in clause 6 is successful.

NOTE — The production and the distribution of keys fall outside the scope of this International Standard.

## 5 Signature process

Figure 1 summarizes the signature process.

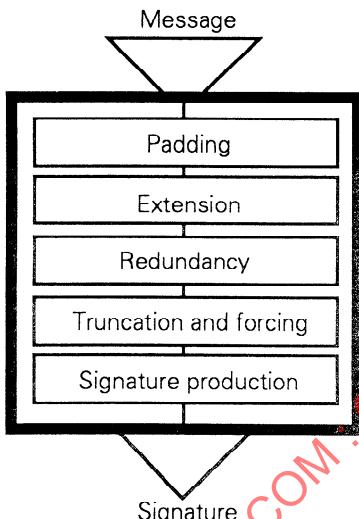


Figure 1 — Signature process

NOTE — A good implementation of the signature process should physically protect the operations in such a way that there is no direct access to the signature function under control of the secret signature key.

### 5.1 Padding

The message is a string of bits. This string of bits is padded to the left by 0 to 7 zeroes so as to obtain a string of  $z$  bytes. Index  $r$ , to be used later on, is the number of padded zeroes plus one. Index  $r$  is thus valued from 1 to 8.

Consequently, in the padded message denoted by  $MP$ , the  $8z+1-r$  least significant bits are information bearing.

$$MP = m_z \parallel m_{z-1} \parallel \dots \parallel m_2 \parallel m_1$$

$$m_z = (r-1 \text{ padded zeroes}) \parallel (9-r \text{ information bits})$$

Number  $z$  multiplied by sixteen shall be less than or equal to number  $k_s+3$ . Consequently, the number of bits of the message to be signed shall be at most 8 times the largest integer less than or equal to  $(k_s+3)/16$ .

### 5.2 Extension

Number  $t$ , to be used later on, is the least integer such that a string of  $2t$  bytes includes at least  $k_s-1$  bits.

The extended message  $ME$  is obtained by repeating the  $z$  bytes of  $MP$ , as many times as necessary, in order and concatenated to the left, until forming a string of  $t$  bytes.

For  $i$  valued from 1 to  $t$  and  $j$  equal to  $i-1 \pmod z$  plus one ( $j$  is therefore valued from 1 to  $z$ ), the  $i$ -th byte of  $ME$  equals the  $j$ -th byte of  $MP$ .

$$ME = \dots m_z \parallel \dots m_2 \parallel m_1$$

$\longleftrightarrow$   $t$  bytes  $\longrightarrow$

NOTE — Number  $z$  is less than or equal to number  $t$ . The equality may occur only if  $k_s$  is congruent to 13, 14, 15, 0 or 1 mod 16.

### 5.3 Redundancy

The extended message with redundancy  $MR$  is obtained by interleaving the  $t$  bytes of  $ME$  in odd positions and  $t$  bytes of redundancy in even positions. Altered by index  $r$ , the least significant nibble of the  $2z$ -th byte of  $MR$  codes the message length by its value and its position.

For  $i$  valued from 1 to  $t$ ,

- the  $(2i-1)$ -th byte of  $MR$  equals the  $i$ -th byte of  $ME$ ;
- the  $2i$ -th byte of  $MR$  equals the image of the  $i$ -th byte of  $ME$  according to the shadow  $S$  specified in table 1, except for the  $2z$ -th byte of  $MR$  which equals the exclusive or of index  $r$  with the shadow of the  $z$ -th byte of  $ME$ .

$$MR = \dots S(m_z) \oplus r \parallel m_z \parallel \dots S(m_2) \parallel m_2 \parallel S(m_1) \parallel m_1$$

$\longleftrightarrow$   $2t$  bytes  $\longrightarrow$

NOTE — The computation of the  $2t$  bytes of  $MR$  ( $mr_{2i}$  to  $mr_1$ ) from the  $z$  bytes of  $MP$  ( $mp_z$  to  $mp_1$ ) is performed by applying successively the following three formulae for  $i$  valued from 1 to  $t$ .

$$j := (i-1 \bmod z)+1; \quad mr_{2i-1} := mp_j; \quad mr_{2i} := S(mp_j)$$

Finally, the  $2z$ -th byte is altered by index  $r$ .

$$mr_{2z} := r \oplus mr_{2z}$$

### 5.4 Truncation and forcing

The intermediate integer  $IR$  is coded by a string of  $k_s$  bits where the most significant bit is valued to 1 and where the  $k_s-1$  least significant bits are those of  $MR$ , except for the least significant byte which is replaced. If  $\mu_2 \parallel \mu_1$  is the least significant byte of  $MR$ , then the least significant byte of  $IR$  shall be  $\mu_1 \parallel 6$ .

### 5.5 Signature production

The signature  $\Sigma$  is obtained as a string of  $k_s$  bits by applying to  $IR$  the signature function under control of the secret signature key.

$$\Sigma = \text{Sign}(IR)$$

Table 1 — Permutation  $\Pi$  and shadow  $S$ 

$\mu$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\Pi(\mu)$	E	3	5	8	9	4	2	F	0	D	B	6	7	A	C	1

If nibble  $\mu$  consists of the bits  $a_4 a_3 a_2 a_1$ , then under the permutation  $\Pi$ , its image denoted by  $\Pi(\mu)$  consists of the bits  $a_4 \oplus a_2 \oplus a_1 \oplus 1; a_4 \oplus a_3 \oplus a_1 \oplus 1; a_4 \oplus a_3 \oplus a_2 \oplus 1; a_3 \oplus a_2 \oplus a_1$ .

If byte  $m$  consists of the nibbles  $\mu_2 \mu_1$ , then under the shadow  $S$ , its image denoted by  $S(m)$  consists of the nibbles  $\Pi(\mu_2) \Pi(\mu_1)$ .

## 6 Verification process

Figure 2 summarizes the verification process.

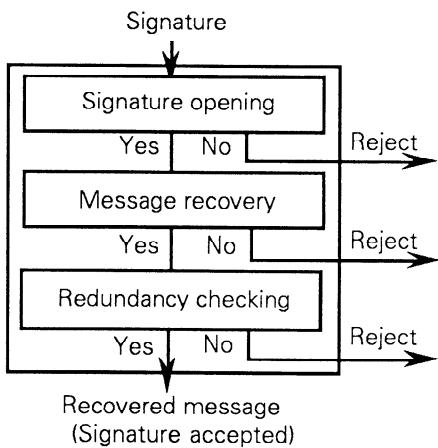


Figure 2 — Verification process

### 6.1 Signature opening

The signature  $\Sigma$  is transformed into the recovered intermediate integer  $IR'$  by applying to  $\Sigma$  the verification function under control of the public verification key.

$$IR' = \text{Verif}(\Sigma)$$

The signature  $\Sigma$  shall be rejected if  $IR'$  is not a string of  $k_s$  bits where the most significant bit is valued to 1 and where the least significant nibble is valued to 6.

### 6.2 Message recovery

The recovered message with redundancy  $MR'$  is the string of  $2t$  bytes where the  $1-k_s$  (mod 16) most significant bits are null and where the  $k_s-1$  least significant bits are those of  $IR'$ , except for the least significant byte which is replaced. According to the permutation  $\Pi$

specified in table 1, if  $\mu_4 \parallel \mu_3 \parallel \mu_2 \parallel 6$  are the four least significant nibbles of  $IR'$ , then the least significant byte of  $MR'$  shall be  $\Pi^{-1}(\mu_4) \parallel \mu_2$ .

$$MR' = m_{2t} \parallel m_{2t-1} \parallel \dots \parallel m_2 \parallel m_1$$

NOTE — The strings  $MR$  and  $MR'$  may be unequal. The string  $MR'$  consists of the  $k_s-1$  least significant bits of  $MR$  padded by 0 to 15 zeroes in the most significant bits.

From the  $2t$  bytes of  $MR'$ ,  $t$  sums are computed. According to the shadow  $S$  specified in table 1, the  $i$ -th sum equals the exclusive-or of the  $2i$ -th byte with the shadow of the  $(2i-1)$ -th byte.

$$m_{2i} \oplus S(m_{2i-1})$$

The signature  $\Sigma$  shall be rejected if the  $t$  sums are null.

Number  $z$  is recovered as the position of the first non-null sum. The recovered padded message  $MP'$  is the string of the  $z$  least significant bytes in odd positions in  $MR'$ .

$$MP' = m_{2z-1} \parallel m_{2z-3} \parallel \dots \parallel m_{2i-1} \parallel \dots \parallel m_3 \parallel m_1$$

Index  $r$  is recovered as the value of the least significant nibble of the first non-null sum.

The signature  $\Sigma$  shall be rejected if index  $r$  is not valued from 1 to 8, and also if the  $r-1$  most significant bits of  $MP'$  are not all null.

$$m_{2z-1} = (r-1 \text{ padded zeroes}) \parallel (9-r \text{ information bits})$$

The message is recovered as the string of the  $8z+1-r$  least significant bits of  $MP'$ .

### 6.3 Redundancy checking

The signature  $\Sigma$  shall be accepted if and only if the  $k_s-1$  least significant bits of  $MR'$  are equal to the  $k_s-1$  least significant bits of another extended message with redundancy computed from the recovered padded message  $MP'$  according to 5.2 and 5.3.

## Annex A

(informative)

### Example of a public-key system for digital signature

#### A.1 Definitions

**Modulus**: Integer constructed as the product of two primes.

**Public verification key**: Modulus and verification exponent.

**Secret signature key**: Signature exponent.

#### A.2 Symbols and abbreviations

<i>RR</i>	Representative element
<i>IS</i>	Resulting integer
<i>n</i>	Modulus
<i>k</i>	Length of the modulus in bits
<i>p, q</i>	Prime factors of the modulus
<i>v</i>	Verification exponent
<i>s</i>	Signature exponent
<i>lcm(a, b)</i>	Least common multiple of integers <i>a</i> and <i>b</i>
$(a n)$	Jacobi symbol of <i>a</i> with respect to <i>n</i>

NOTE — Let *p* be an odd prime, and let *a* be a positive integer. The Legendre symbol of integer *a* with respect to prime *p* is defined by the following formula.

$$(a|p) = a^{(p-1)/2} \bmod p$$

When integer *a* is not a multiple of *p*, then the Legendre symbol of integer *a* with respect to prime *p* is valued to either +1 or -1 depending on whether integer *a* is or is not a square modulo *p*.

The Legendre symbol of multiples of *p* with respect to prime *p* is null.

Let *n* be an odd positive integer, and let *a* be a positive integer. The Jacobi symbol of integer *a* with respect to integer *n* is the product of the Legendre symbols of integer *a* with respect to the prime factors of *n*.

Therefore if *n* = *p q*, then  $(a|n) = (a|p)(a|q)$ .

The Jacobi symbol of any integer *a* with respect to any integer *n* may be efficiently computed without the prime factors of *n*.

#### A.3 Key production

##### A.3.1 Public verification exponent

Each signing entity shall select a positive integer *v* as its public verification exponent.

The public verification exponent may be standardized in specific applications.

NOTE — Values 2 and 3 may have some practical advantages.

##### A.3.2 Secret prime factors and public modulus

Each signing entity shall secretly and randomly select two distinct odd primes *p* and *q* subject to the following conditions.

— If *v* is odd, then *p*-1 and *q*-1 shall be coprime to *v*.

— If *v* is even, then  $(p-1)/2$  and  $(q-1)/2$  shall be coprime to *v*. Moreover, *p* and *q* shall not be congruent to each other mod 8.

The public modulus *n* is the product of the secret prime factors *p* and *q*.

$$n = p q$$

The length of the modulus is *k*. Number *k* shall equal  $k_s + 1$ .

##### NOTES

1 Some additional conditions on the choice of primes may well be taken into account in order to deter factorization of the modulus.

2 Some forms of the modulus simplify the modulo reduction and need less table storage. These forms are

$$F_{x, y, -} : n = 2^{64x} - c \quad \text{of length: } k = 64x \text{ bits,}$$

$$F_{x, y, +} : n = 2^{64x} + c \quad \text{of length: } k = 64x + 1 \text{ bits,}$$

$$\text{where: } 1 \leq y \leq 2x \text{ and } c < 2^{64x-8y} < 2c.$$

In the negative forms, all the bits of the *y* most significant bytes are valued to one, up to a quarter of the length of the modulus.

In the positive forms, after a single most significant bit valued to one, all the bits of the *y* most significant bytes are valued to zero, up to a quarter of the length of the modulus.

##### A.3.3 Secret signature exponent

The secret signature exponent is the least positive integer *s* such that  $sv-1$  is a multiple of

—  $\text{lcm}(p-1, q-1)$  if *v* is odd;

—  $\frac{1}{2}\text{lcm}(p-1, q-1)$  if *v* is even.

#### A.4 Signature function

The intermediate integer  $IR$  is a string of  $k-1$  bits computed as described in 5.4.

The representative element of  $IR$  with respect to  $n$  is denoted by  $RR$ .

- If  $v$  is odd, then  $RR$  is  $IR$ .
- If  $v$  is even and if  $(IR \mid n) = +1$ , then  $RR$  is  $IR$ .
- If  $v$  is even and if  $(IR \mid n) = -1$ , then  $RR$  is  $IR/2$ .

NOTE — If  $v$  is even, then the Jacobi symbol of  $RR$  with respect to  $n$  is forced to  $+1$ .

$RR$  shall be raised to the power  $s$  modulo  $n$ . The signature  $\Sigma$  is either the result or its complement to  $n$ , the least one.

$$\Sigma = \min \{ RR^s \bmod n, n - (RR^s \bmod n) \}$$

This defines the signature function "Sign".

$$\Sigma = \text{Sign}(IR)$$

#### A.5 Verification function

The signature  $\Sigma$  is a positive integer less than  $n/2$  which shall be raised to the power  $v$  modulo  $n$  for obtaining the resulting integer  $IS$ .

The recovered intermediate integer  $IR'$  is then defined by the following decoding.

- If  $IS$  is congruent to  $6 \bmod 16$ , then  $IR'$  is  $IS$ .
- If  $n-IS$  is congruent to  $6 \bmod 16$ , then  $IR'$  is  $n-IS$ .

Moreover, when  $v$  is even,

- if  $IS$  is congruent to  $3 \bmod 8$ , then  $IR'$  is  $2IS$ ;
- if  $n-IS$  is congruent to  $3 \bmod 8$ , then  $IR'$  is  $2(n-IS)$ .

The signature  $\Sigma$  shall be rejected in all the other cases, and also if  $IR'$  does not lie in the range from  $2^{k-2}$  to  $2^{k-1}-1$ .

This defines the verification function "Verif".

$$IR' = \text{Verif}(\Sigma)$$

## Annex B

(informative)

### Illustrative examples related to annex A

The hexadecimal notation is used.

#### B.1 Examples with public exponent three

##### B.1.1 Key production

The public verification exponent  $v$  is 3.

Therefore the secret prime factors are both congruent to  $2 \pmod{3}$ .

$p = \begin{array}{llllll} \text{BA09106C} & 754EB6FE & \text{BBC21479} & 9FF1B8DE \\ 1B4CBB7A & 7A782B15 & 7C1BC152 & 90A1A3AB \end{array}$

$q = \begin{array}{llllll} 1\ 6046EB39 & E03BEAB6 & 21D03C08 & B8AE6B66 \\ \text{CFF955B6} & 4B4F48B7 & \text{EE152A32} & 6BF8CB25 \end{array}$

The public modulus  $n$  of 513 bits is of the form  $2^{512} + c$ , with  $2c > 2^{384} > c$  (form  $F_x, y, +$  with  $x = 8$  and  $y = 16$ ).

$n = p \cdot q = \begin{array}{llllll} 1\ 00000000 & 00000000 & 00000000 & 00000000 \\ \text{BBA2D15D} & \text{BB303C8A} & 21C5EBBC & \text{BAE52B71} \\ 25087920 & \text{DD7CDF35} & 8EA119FD & 66FB0640 \\ 12EC8CE6 & 92F0A0B8 & \text{E8321B04} & 1ACD40B7 \end{array}$

The secret signature exponent  $s$  is  $(n-p-q+3)/6$ .

$s = \begin{array}{llllll} 2\text{AAAAAAA} & \text{AAAAAAA} & \text{AAAAAAA} & \text{AAAAAAA} \\ \text{C9F0783A} & 49DD5F6C & 5AF651F4 & \text{C9D0DC92} \\ 81C96A3F & 16A85F95 & 72D7CC3F & 2D0F25A9 \\ \text{DBF1149E} & \text{4CDC3227} & 3FAADD3F & \text{DA5DCDA7} \end{array}$

##### B.1.2 Length of the variables

Number  $z$  is a positive integer less than or equal to  $k+2$  divided by 16. Number  $t$  is the largest integer less than or equal to  $k+13$  divided by 16.

Consequently, when number  $k$  is 513,

- number  $z$  is valued from 1 to 32, the messages to be signed are strings of 1 to 256 bits, and the padded messages  $MP$  and  $MP'$  are strings of 1 to 32 bytes;
- number  $t$  is 32, the extended messages  $ME$  are strings of 32 bytes, and the messages with redundancy  $MR$  and  $MR'$  are strings of 64 bytes.

Moreover, the intermediate integers  $IR$  and  $IR'$  and the signatures  $\Sigma$  are strings of 512 bits ( $k-1$  bits).

##### B.1.3 Example 1

This example illustrates padding, extension and truncation for signing a message of 100 bits.

$C\ \text{BBAA}\ 9988\ 7766\ 5544\ 3322\ 1100$

##### Signature process

After padding four zeroes to the left, the padded message  $MP$  is a string of 13 bytes. Therefore  $z=13$  and  $r=5$ .

$MP = \begin{array}{llllll} 0\ \text{C} & \text{BBAA}\ 9988 & 7766\ 5544 & 3322\ 1100 \end{array}$

The extended message  $ME$  results by repeating the 13 successive bytes of  $MP$ , in order and concatenated to the left, until obtaining a string of 32 bytes.

$ME = \begin{array}{llllll} 55443322 & 1100\ 0\text{C}\ BB & AA\ 9988\ 77 & 66554433 \\ 221100\ 0\text{C} & \text{BBAA}\ 9988 & 7766\ 5544 & 3322\ 1100 \end{array}$

The extended message with redundancy  $MR$  is a string of 64 bytes obtained by interleaving the 32 bytes of  $ME$  and 32 bytes of redundancy. An alteration of the 26-th byte (E2) codes the message border.

$MR = \begin{array}{llllll} 44559944 & 88335522 & 3311EE00 & E70C66BB \\ \text{BBAADD99} & 0088FF77 & 22664455 & 99448833 \\ 55223311 & \text{EE00}\ 0\text{E}\ 20\text{C} & 66BBBAA & \text{DD990088} \\ \text{FF772266} & 44559944 & 88335522 & 3311EE00 \end{array}$

The intermediate integer  $IR$  results from  $MR$  by truncating to 511 bits, by padding to the left one bit valued to 1 and by replacing the least significant byte:  $\mu_2 \parallel \mu_1 = 00$  is replaced by  $\mu_1 \parallel 6 = 06$ .

Because  $v$  is odd, the representative element  $RR$  is  $IR$ .

$RR = IR = \begin{array}{llllll} \text{C4559944} & 88335522 & 3311EE00 & E70C66BB \\ \text{BBAADD99} & 0088FF77 & 22664455 & 99448833 \\ 55223311 & \text{EE00}\ 0\text{E}\ 20\text{C} & 66BBBAA & \text{DD990088} \\ \text{FF772266} & 44559944 & 88335522 & 3311EE00 \end{array}$

$RR$  is raised to the power  $s$  modulo  $n$ . The signature  $\Sigma$  is here the complement to  $n$  of the result.

$\Sigma = \begin{array}{llllll} 309F873D & 8DED8379 & 490F6097 & \text{EAFDABC} \\ 137D3EBF & D8F25AB5 & F138D56A & 719CDC52 \\ 6BDD022E & A65DABAB & 920A8101 & 3A85D092 \\ \text{E04D3E42} & 1CAAB717 & \text{C90D89EA} & 45A8D23A \end{array}$

## Verification process

The signature  $\Sigma$  is less than  $n/2$ . The resulting integer  $IS$  is obtained by raising  $\Sigma$  to the power 3 modulo  $n$ .

$IS = \begin{array}{llllll} 3BAA66BB & 77CCAADD & CCEE11FF & 18F39944 \\ FFF7F3C4 & BAA73D12 & FF5FA767 & 21A0A33D \\ CFE6460E & EF7BFD29 & 27E55E52 & 896205B7 \\ 13756A80 & 4E9B0774 & 5FFEC5E1 & E7BB52\mathbf{B1} \end{array}$

The intermediate integers are strings of 512 bits where the most significant bit is valued to 1 and the least significant nibble is valued to 6. Because  $n$  is here congruent to  $7 \bmod 16$  and  $IS$  to  $1 \bmod 16$ , the recovered intermediate integer  $IR'$  is  $n-IS$ .

$IR' = n-IS = \begin{array}{llllll} C4559944 & 88335522 & 3311EE00 & E70C66BB \\ BBAADD99 & 0088FF77 & 22664455 & 99448833 \\ 55223311 & EE00E20C & 66BBBBAA & DD990088 \\ FF772266 & 44559944 & 88335522 & 3311EE\mathbf{06} \end{array}$

The recovered message with redundancy  $MR'$  is here the string of 64 bytes where a padded zero is followed by the 511 least significant bits of  $IR'$ , except for the least significant byte; according to the permutation  $\Pi$  stating  $\Pi(0) = E$ ,  $EE06$  denoted by  $\mu_4 \parallel \mu_3 \parallel \mu_2 \parallel 6$  is replaced by  $\mu_4 \parallel \mu_3 \parallel \Pi^{-1}(\mu_4) \parallel \mu_2$  valued to  $EE00$ .

$MR' = \begin{array}{llllll} 44559944 & 88335522 & 3311EE00 & E70C66BB \\ BBAADD99 & 0088FF77 & 22664455 & 99448833 \\ 55223311 & \mathbf{EE00E20C} & 66BBBBAA & DD990088 \\ FF772266 & 44559944 & 88335522 & 3311EE\mathbf{00} \end{array}$

The first non-null sum is the 13-th sum valued to 5. Thus  $z=13$  and  $r=5$ . The recovered padded message  $MP'$  is the string of the 13 bytes of  $MR'$  in the least significant odd positions.

$MP' = \begin{array}{llllll} 0C & BBA9988 & 77665544 & 33221100 & \dots \end{array}$

The four most significant bits ( $r-1=4$ ) of  $MP'$  are null. The message itself is recovered as the string of the least significant 100 bits ( $8z+1-r=100$ ) of  $MP'$ .

C BAA 9988 7766 5544 3322 1100

The signature is accepted because the 511 least significant bits of the recovered message with redundancy  $MR'$  are recovered in the extended message with redundancy computed from  $MP'$ , exactly as  $MR$  from  $MP$ .

### B.1.4 Example 2

This example illustrates a simpler case: a 256-bit message is neither padded nor extended with a 513-bit modulus.

FEDC BA98 7654 3210 FEDC BA98 7654 3210  
FEDC BA98 7654 3210 FEDC BA98 7654 3210

### Signature process

The message is a string of 256 bits, coded over exactly 32 bytes. Therefore  $z$  is 32 and  $r$  is 1. The message equals the padded message  $MP$  and the extended message  $ME$ .

$ME = MP = \begin{array}{llllll} FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \\ FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \end{array}$

The extended message with redundancy  $MR$  is a string of 64 bytes.

$MR = \begin{array}{llllll} 1DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \end{array}$

The intermediate integer  $IR$  results from  $MR$  by truncating to 511 bits, by padding to the left one bit valued to 1 and by replacing the least significant byte.

Because  $v$  is odd, the representative element  $RR$  is  $IR$ .

$RR = IR = \begin{array}{llllll} 9DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E06 \end{array}$

$RR$  is raised to the power  $s$  modulo  $n$ . The signature  $\Sigma$  is here the result.

$\Sigma = \begin{array}{llllll} 319BB9BE & CB49F3ED & 1BCA26D0 & FCF09B0B \\ 0A508E4D & 0BD43B35 & 0F959B72 & CD25B3AF \\ 47D608FD & CD248EAD & A74FBE19 & 990DBEB9 \\ BF0DA4B4 & E1200243 & A14E5CAB & 3F7E610C \end{array}$

### Verification process

The signature  $\Sigma$  is less than  $n/2$ . The resulting integer  $IS$  is obtained by raising  $\Sigma$  to the power 3 modulo  $n$ .

Because  $IS$  is here congruent to  $1 \bmod 16$ , the recovered intermediate integer  $IR'$  is here  $IS$ .

$IR' = IS = \begin{array}{llllll} 9DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E06 \end{array}$

The recovered message with redundancy  $MR'$  is here the string of 64 bytes where a padded zero is followed by the 511 least significant bits of  $IR'$ , except for the least significant byte; according to the permutation  $\Pi$  stating  $\Pi(1) = 3$ ,  $3E06$  denoted by  $\mu_4 \parallel \mu_3 \parallel \mu_2 \parallel 6$  is replaced by  $\mu_4 \parallel \mu_3 \parallel \Pi^{-1}(\mu_4) \parallel \mu_2$  valued to  $3E10$ .

$MR' = \begin{array}{llllll} 1DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \end{array}$

The first non-null sum is the 32-nd sum valued to 1. Thus  $z=32$  and  $r=1$ . The recovered padded message  $MP'$  is the string of the 32 bytes of  $MR'$  in odd positions.

$MP' = \begin{array}{llllll} FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \\ FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \end{array}$

The recovered message is a string of 256 bits.

FEDC BA98 7654 3210 FEDC BA98 7654 3210  
FEDC BA98 7654 3210 FEDC BA98 7654 3210

The signature is accepted because the 511 least significant bits of the recovered message with redundancy  $MR'$  are recovered in the extended message with redundancy computed from  $MP'$ , exactly as  $MR$  from  $MP$ .

## B.2 Another example with public exponent three

### B.2.1 Key production

The public verification exponent  $v$  is 3.

Therefore the secret prime factors are both congruent to 2 mod 3.

$p = \begin{array}{cccc} 461908C5 & 405B7952 & F69864C3 & B0683002 \\ 5650303D & 5297A4BD & 2F549A9D & 37CFE027 \end{array}$

$q = \begin{array}{cccc} 3 A6EC260F & 3E2E0B2C & 106C5164 & 6D471D9E \\ 04783176 & 27010818 & E54CC26F & 7C0C892B \end{array}$

The public modulus  $n$  of 512 bits is of the form  $2^{512} - c$ ,  
with  $2c > 2^{488} > c$  (form  $F_x, y$ , with  $x = 8$  and  $y = 3$ ).

$n = p \cdot q = \begin{array}{cccc} \text{FFFFFFFFFF7F} & \text{A27087C3} & \text{5EBEAD78} & \text{412D2BDF} \\ \text{FE0301ED} & \text{D494DF13} & \text{458974EA} & \text{89B36470} \\ \text{8F7D0F5A} & \text{00A50779} & \text{DDF9F7D4} & \text{CB80B889} \\ \text{1324DA25} & \text{1A860C4E} & \text{C9EF2881} & \text{04B3858D} \end{array}$

The secret signature exponent  $s$  is  $(n-p-q+3)/6$ .

$s = \begin{array}{cccc} 2AAAAAA95 & 45BD6BF5 & E51FC794 & 0ADCDCA5 \\ 55008052 & 4E18CFD8 & 8B96E8D1 & C19DE612 \\ 1B13FAC0 & EB0495D4 & 7928E047 & 724D91D1 \\ 740F6968 & 457CE53E & C8E24C93 & 62CE84B5 \end{array}$

### B.2.2 Length of the variables

Because number  $k$  is 512,

- number  $z$  is valued from 1 to 32, the messages to be signed are strings of 1 to 256 bits, and the padded messages  $MP$  and  $MP'$  are strings of 1 to 32 bytes;
- number  $t$  is 32, the extended messages  $ME$  are strings of 32 bytes, and the messages with redundancy  $MR$  and  $MR'$  are strings of 64 bytes.

Moreover, the intermediate integers  $IR$  and  $IR'$  and the signatures  $\Sigma$  are strings of 511 bits ( $k-1$  bits).

### B.2.3 Example 3

This example illustrates padding, extension and truncation for signing a message of 100 bits.

1 1223 3445 5667 7889 9AAB BCCD

#### Signature process

After padding four zeroes to the left, the padded message  $MP$  is a string of 13 bytes. Therefore  $z=13$  and  $r=5$ .

$MP = \begin{array}{ccccccccccccc} 01 & 12233445 & 56677889 & 9AAB & BCCD \end{array}$

The extended message  $ME$  results by repeating the 13 successive bytes of  $MP$ , in order and concatenated to the left, until obtaining a string of 32 bytes.

$ME = \begin{array}{cccccccccccccccc} 78899AAB & BCCD0112 & 23344556 & 6778899A \\ ABBCCD01 & 12233445 & 56677889 & 9AABBCCD \end{array}$

The extended message with redundancy  $MR$  is a string of 64 bytes obtained by interleaving the 32 bytes of  $ME$  and 32 bytes of redundancy. An alteration of the 26-th byte (E6) codes the message border.

$MR = \begin{array}{cccccccccccccccccccccccc} F0780D89 & DB9AB6AB & 67BC7ACD & E3013512 \\ 58238934 & 94454256 & 2F67F078 & 0D89DB9A \\ B6AB67BC & 7ACDE601 & 35125823 & 89349445 \\ 42562F67 & F0780D89 & DB9AB6AB & 67BC7ACD \end{array}$

The intermediate integer  $IR$  results from  $MR$  by truncating to 510 bits, by padding to the left one bit valued to 1 and by replacing the least significant byte:  $\mu_2 \parallel \mu_1 = CD$  is replaced by  $\mu_1 \parallel 6 = D6$ .

Because  $v$  is odd, the representative element  $RR$  is  $IR$ .

$RR = IR = \begin{array}{cccccccccccccccccccccccc} 70780D89 & DB9AB6AB & 67BC7ACD & E3013512 \\ 58238934 & 94454256 & 2F67F078 & 0D89DB9A \\ B6AB67BC & 7ACDE601 & 35125823 & 89349445 \\ 42562F67 & F0780D89 & DB9AB6AB & 67BC7AD6 \end{array}$

$RR$  is raised to the power  $s$  modulo  $n$ . The signature  $\Sigma$  is here the complement to  $n$  of the result.

$\Sigma = \begin{array}{cccccccccccccccccccccccc} 58E59FFB & 4B1FB1BC & DBF8D1FE & 9AFA3730 \\ C78A318A & 1134F579 & 1B7313D4 & 80F07AC \\ 319B068E & DF8F2129 & 45CB09CF & 33DF30AC \\ E54F4A06 & 3FCCA0B7 & 32F4B662 & DC4E2454 \end{array}$

#### Verification process

The signature  $\Sigma$  is less than  $n/2$ . The resulting integer  $IS$  is obtained by raising  $\Sigma$  to the power 3 modulo  $n$ .

$IS = \begin{array}{cccccccccccccccccccccccc} 8F87F1F5 & C6D5D117 & F70232AA & 5E2BF6CD \\ A5DF78B9 & 404F9CBD & 16218472 & 7C2998D5 \\ D8D1A79D & 85D72178 & A8E79FB1 & 424C2443 \\ D0CEAABD & 2A0DFEC4 & EE5471D5 & 9CF70AB7 \end{array}$

The intermediate integers are strings of 511 bits where the most significant bit is valued to 1 and the least significant nibble is valued to 6. Because  $n$  is here congruent to 13 mod 16 and  $IS$  to 7 mod 16, the recovered intermediate integer  $IR'$  is  $n-IS$ .

$IR' = n-IS = \begin{array}{cccccccccccccccccccccccc} 70780D89 & DB9AB6AB & 67BC7ACD & E3013512 \\ 58238934 & 94454256 & 2F67F078 & 0D89DB9A \\ B6AB67BC & 7ACDE601 & 35125823 & 89349445 \\ 42562F67 & F0780D89 & DB9AB6AB & 67BC7AD6 \end{array}$

The recovered message with redundancy  $MR'$  is here the string of 64 bytes where two padded zeroes are followed by the 510 least significant bits of  $IR'$ , except for the least significant byte; according to the permutation  $\Pi$  stating  $\Pi(C) = 7$ , 7AD6 denoted by  $\mu_4 \parallel \mu_3 \parallel \mu_2 \parallel 6$  is replaced by  $\mu_4 \parallel \mu_3 \parallel \Pi^{-1}(\mu_4) \parallel \mu_2$  valued to 7ACD.

$MR' = \begin{array}{cccccccccccccccccccccccc} 30780D89 & DB9AB6AB & 67BC7ACD & E3013512 \\ 58238934 & 94454256 & 2F67F078 & 0D89DB9A \\ B6AB67BC & 7ACDE601 & 35125823 & 89349445 \\ 42562F67 & F0780D89 & DB9AB6AB & 67BC7ACD \end{array}$

The first non-null sum is the 13-th sum valued to 5. Thus  $z=13$  and  $r=5$ . The recovered padded message  $MP'$  is the string of the 13 bytes of  $MR'$  in the least significant odd positions.

$MP' = \begin{array}{ccccccccccccc} 01 & 12233445 & 56677889 & 9AABBCCD \end{array}$

The four most significant bits ( $r-1=4$ ) of  $MP'$  are null. The message itself is recovered as the string of the least significant 100 bits ( $8z+1-r=100$ ) of  $MP'$ .

1 1223 3445 5667 7889 9AAB BCCD

The signature is accepted because the 510 least significant bits of the recovered message with redundancy  $MR'$  are recovered in the extended message with redundancy computed from  $MP'$ , exactly as  $MR$  from  $MP$ .

### B.3 Examples with public exponent two

#### B.3.1 Key production

The public verification exponent  $v$  is 2.

Therefore, one secret prime factor is congruent to 3 mod 8 and the other one is congruent to 7 mod 8.

$p = \begin{array}{cccccc} 867EA672 & E46B2B0A & 35F2F2F2 & 719A1F3C \\ 7EA05947 & 2B9DAE51 & A1730A28 & 2CDBBE3 \end{array}$

$q = \begin{array}{cccccc} 1 & E7468E3C & 4869473F & 094E7406 & 60B04CB4 \\ 8E47FB50 & 196544DC & C81D4492 & 8301850F & \end{array}$

The public modulus  $n$  of 513 bits is of the form  $2^{512} + c$ , with  $2c > 2^{384} > c$  (form  $F_x, y, +$  with  $x = 8$  and  $y = 16$ ).

$n = p \cdot q = \begin{array}{cccccc} 1 & 00000000 & 00000000 & 00000000 & 00000000 \\ 97518F6A & D742E4E3 & A1EDC7F6 & CB0F2226 \\ F1343952 & 4E5466C2 & D596A9F9 & 760FAD26 \\ 743E5D43 & D9AAA91E & F0368F22 & B87DF14D \end{array}$

The secret signature exponent  $s$  is  $(n-p-q+5)/8$ .

$s = \begin{array}{cccccc} 20000000 & 00000000 & 00000000 & 00000000 \\ 12EA31ED & 5AE85C9C & 743DB8FE & D961E444 \\ 906DE094 & 642FFE8F & 32CAA860 & 1478A826 \\ ACEAC115 & 9294F6BE & 10D4C80D & 0113D60C \end{array}$

#### B.3.2 Length of the variables

Because number  $k$  is 513,

- number  $z$  is valued from 1 to 32, the messages to be signed are strings of 1 to 256 bits, and the padded messages  $MP$  and  $MP'$  are strings of 1 to 32 bytes;
- number  $t$  is 32, the extended messages  $ME$  are strings of 32 bytes, and the messages with redundancy  $MR$  and  $MR'$  are strings of 64 bytes.

Moreover, the intermediate integers  $IR$  and  $IR'$  and the signatures  $\Sigma$  are strings of 512 bits ( $k-1$  bits).

#### B.3.3 Example 4

This example illustrates the signature of a message of 256 bits with forcing the Jacobi symbol.

F123 E123 D123 C123 B123 A123 9123 8123  
7123 6123 5123 4123 3123 2123 1123 0123

#### Signature process

The message is a string of 256 bits, coded over exactly 32 bytes. Therefore  $z$  is 32 and  $r$  is 1. The message equals the padded message  $MP$  and the extended message  $ME$ .

$ME = MP = \begin{array}{cccccc} F123E123 & D123C123 & B123A123 & 91238123 \\ 71236123 & 51234123 & 31232123 & 11230123 \end{array}$

The extended message with redundancy  $MR$  is a string of 64 bytes.

$MR = \begin{array}{cccccc} 12F15823 & C3E15823 & A3D15823 & 73C15823 \\ 63B15823 & B3A15823 & D3915823 & 03815823 \\ F3715823 & 23615823 & 43515823 & 93415823 \\ 83315823 & 53215823 & 33115823 & E3015823 \end{array}$

The intermediate integer  $IR$  results by truncating  $MR$  to 511 bits, by padding to the left one bit valued to 1 and by replacing the least significant byte.

$IR = \begin{array}{cccccc} 92F15823 & C3E15823 & A3D15823 & 73C15823 \\ 63B15823 & B3A15823 & D3915823 & 03815823 \\ F3715823 & 23615823 & 43515823 & 93415823 \\ 83315823 & 53215823 & 33115823 & E3015826 \end{array}$

Because the Jacobi symbol of  $IR$  with respect to  $n$  is  $-1$ , the representative element  $RR$  is  $IR/2$ .

$RR = IR/2 = \begin{array}{cccccc} 4978AC11 & E1F0AC11 & D1E8AC11 & B9E0AC11 \\ B1D8AC11 & D9D0AC11 & E9C8AC11 & 81C0AC11 \\ F9B8AC11 & 91B0AC11 & A1A8AC11 & C9A0AC11 \\ C198AC11 & A990AC11 & 9988AC11 & F180AC1B \end{array}$

$RR$  is raised to the power  $s$  modulo  $n$ . The signature  $\Sigma$  is here the result.

$\Sigma = \begin{array}{cccccc} 6BA03660 & D7A9001D & 533B01A6 & 05CAF2A \\ 1352E0D7 & 8776623C & 926FF204 & 3B93E12B \\ E7D097AE & 50624815 & 3024E3C1 & 7CFA565D \\ F4F76FF2 & EC19C507 & 9D11C723 & F0CE5071 \end{array}$

#### Verification process

The signature  $\Sigma$  is less than  $n/2$ . The resulting integer  $IS$  is obtained by squaring  $\Sigma$  mod  $n$ .

$IS = \begin{array}{cccccc} 0 & 4978AC11 & E1F0AC11 & D1E8AC11 & B9E0AC11 \\ B1D8AC11 & D9D0AC11 & E9C8AC11 & 81C0AC11 \\ F9B8AC11 & 91B0AC11 & A1A8AC11 & C9A0AC11 \\ C198AC11 & A990AC11 & 9988AC11 & F180AC1B \end{array}$

Because  $IS$  is here congruent to 11 mod 16, the recovered intermediate integer  $IR'$  is here  $2IS$ .

$IR' = 2IS = \begin{array}{cccccc} 92F15823 & C3E15823 & A3D15823 & 73C15823 \\ 63B15823 & B3A15823 & D3915823 & 03815823 \\ F3715823 & 23615823 & 43515823 & 93415823 \\ 83315823 & 53215823 & 33115823 & E3015836 \end{array}$

The recovered message with redundancy  $MR'$  is the string of 64 bytes equal to  $IR'$ , except for the most significant bit forced to 0 and the least significant byte which is replaced ( $\Pi^1(5)=2$ ).

$MR' = \begin{array}{cccccc} 12F15823 & C3E15823 & A3D15823 & 73C15823 \\ 63B15823 & B3A15823 & D3915823 & 03815823 \\ F3715823 & 23615823 & 43515823 & 93415823 \\ 83315823 & 53215823 & 33115823 & E3015823 \end{array}$

The first non-null sum is the 32-nd sum valued to 1. Thus  $z=32$  and  $r=1$ . The recovered padded message  $MP'$  is the string of the 32 bytes of  $MR'$  in odd positions.

$MP' = \begin{array}{ccccccccc} F123E123 & D123C123 & B123A123 & 91238123 \\ 71236123 & 51234123 & 31232123 & 11230123 \end{array}$

The recovered message is a string of 256 bits.

$F123 \ E123 \ D123 \ C123 \ B123 \ A123 \ 9123 \ 8123 \\ 7123 \ 6123 \ 5123 \ 4123 \ 3123 \ 2123 \ 1123 \ 0123$

The signature is accepted because the 511 least significant bits of the recovered message with redundancy  $MR'$  are recovered in the extended message with redundancy computed from  $MP'$ , exactly as  $MR$  from  $MP$ .

### B.3.4 Example 5

This last example illustrates the signature of a message of 256 bits without forcing the Jacobi symbol.

$FEDC \ BA98 \ 7654 \ 3210 \ FEDC \ BA98 \ 7654 \ 3210 \\ FEDC \ BA98 \ 7654 \ 3210 \ FEDC \ BA98 \ 7654 \ 3210$

#### Signature process

The message is a string of 256 bits, coded over exactly 32 bytes. Therefore  $z$  is 32 and  $r$  is 1. The message equals the padded message  $MP$  and the extended message  $ME$ .

$ME = MP = \begin{array}{ccccccccc} FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \\ FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \end{array}$

The extended message with redundancy  $MR$  is a string of 64 bytes.

$MR = \begin{array}{ccccccccc} 1DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \end{array}$

The intermediate integer  $IR$  results by truncating  $MR$  to 511 bits, by padding to the left one bit valued to 1 and by replacing the least significant byte. And because the Jacobi symbol of  $IR$  with respect to  $n$  is +1,  $IR$  is here the representative element  $RR$ .

$RR = IR = \begin{array}{ccccccccc} 9DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E06 \end{array}$

$RR$  is raised to the power  $s$  modulo  $n$ . The result is here the signature  $\Sigma$ .

$\Sigma = \begin{array}{ccccccccc} 28910D1F & 0FC8332A & 63AFE10A & 37848404 \\ 84374DF9 & E0A92347 & DD1966E5 & 976823EC \\ 597A1AEC & 0D24FE71 & 0934D49B & 0CB0412F \\ E8A10CB0 & D39D1C06 & 207B0000 & E9F33021 \end{array}$

#### Verification process

The signature  $\Sigma$  is less than  $n/2$ . The resulting integer  $IS$  is obtained by squaring  $\Sigma \bmod n$ .

And because  $IS$  is here congruent to 6 mod 16,  $IS$  is here the recovered intermediate integer  $IR'$ .

$IR' = IS = \begin{array}{ccccccccc} 9DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E06 \end{array}$

The recovered message with redundancy  $MR'$  is the string of 64 bytes equal to  $IR'$ , except for the most significant bit forced to 0 and the least significant byte which is replaced ( $\Pi^1(3)=1$ ).

$MR' = \begin{array}{ccccccccc} 1DFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \\ 1CFEA7DC & 6BBAD098 & F2764954 & 85323E10 \end{array}$

The first non-null sum is the 32-nd sum valued to 1. Thus  $z=32$  and  $r=1$ .

The recovered padded message  $MP'$  is the string of the 32 bytes of  $MR'$  in odd positions.

$MP' = \begin{array}{ccccccccc} FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \\ FEDCBA98 & 76543210 & FEDCBA98 & 76543210 \end{array}$

The recovered message is a string of 256 bits.

$FEDC \ BA98 \ 7654 \ 3210 \ FEDC \ BA98 \ 7654 \ 3210 \\ FEDC \ BA98 \ 7654 \ 3210 \ FEDC \ BA98 \ 7654 \ 3210$

The signature is accepted because the 511 least significant bits of the recovered message with redundancy  $MR'$  are recovered in the extended message with redundancy computed from  $MP'$ , exactly as  $MR$  from  $MP$ .

## Annex C

(informative)

### Some precautions taken against various potential attacks related to annex A

#### C.1 Legitimate arguments of the secret function

The only legitimate arguments of the function "raising to the power  $s$  modulo  $n$ " are the representative elements.

If  $v$  is odd, any representative element is a string of  $k-1$  bits where the most significant bit is valued to 1 and where the least significant nibble is valued to 6.

If  $v$  is even, then one forces to +1 the Jacobi symbol of the representative elements with respect to modulus  $n$ . And any representative element is a string of

- $k-1$  bits where the most significant bit is valued to 1 and where the least significant nibble is valued to 6 if  $(I|R|n) = +1$ .

- $k-2$  bits where the most significant bit is valued to 1 and where the string of the three least significant bits is valued to 3 if  $(I|R|n) = -1$ .

#### C.2 Elimination of four operations

Owing to the structure of the representative elements, the following four operations are eliminated.

NOTE — These informations are the scope of a communication (see annex D) presented at Eurocrypt '90, a workshop held in Århus, Denmark, 1990-05-21/24.

#### Shift

No string of bits coding a representative element can be shifted into another representative element.

#### Complementation

No string of bits coding a representative element can be complemented into another representative element.

#### Natural multiplication

The natural product (i.e. without involving the use of a modulo reduction) of a constant by a representative element is never another representative element.

#### Natural power

The natural  $v$ -th power (i.e. without involving the use of a modulo reduction) of a constant is never a representative element.

As a matter of fact, an integer congruent to 6 modulo 16 is never a power; and an integer congruent to 3 modulo 8 is never an even power.

## Annex D (informative)

### Bibliography

[1] *Precautions taken against various potential attacks in ISO/IEC 9796, Digital signature scheme giving message recovery*, Louis GUILLOU, Jean-Jacques QUISQUATER, Mike WALKER, Peter LANDROCK, Caroline SHAER, Proceedings of Eurocrypt '90, edited by Ivan DAMGÅRD and published by Springer-Verlag in the series "Lecture Notes in Computer Science", Vol 473, pp 465-473.

IECNORM.COM: Click to view the full PDF of ISO/IEC 9796:1991