INTERNATIONAL STANDARD

ISO 11166-2

First edition
1994-11-15

Banking — Key management by means of asymmetric algorithms —

Part 2:
Approved algorithms using the RSA cryptosystem

Banque — Gestion des clés au moyen d'algorithmes asymétriques —

Partie 2: Algorithmes approuvés utilisant le système de chiffrement RSA

Reference number
ISO 11166-2:1994(E)

Contents

International Organization for Standardization
Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75% approval by the member bodies voting.

International Standard ISO 11166-2 was prepared by Technical Committee ISO/TC 68, *Banking and relating financial services*, Subcommittee 2, *Operations and procedures*.

ISO 11166 consists of the following parts, under the general title *Banking — Key management by means of asymmetric algorithms:*

— *Part 1: Principles, procedures and formats*

— *Part 2: Approved algorithms using the RSA cryptosystem.*

Annex A forms an integral part of this part of ISO 11166. Annexes B and C are for information only.

Introduction

ISO 11166-1 specifies those aspects of banking key management using asymmetric algorithms which are independent of the algorithms employed. Subsequent parts of the standard specify asymmetric algorithms which are approved for use in the procedures of part 1.

This part of ISO 11166 specifies approved algorithms using the RSA cryptosystem. It also provides a link between the algorithm specification and the notation used in part 1 to denote the primitive cryptographic processes employed in banking key management.

It is envisaged that alternative approved algorithms will be specified in further parts of the standard, no single algorithm being mandatory.

The level of security achievable by the use of a cryptographic algorithm depends, among other factors, on parameters defining the algorithm and on the current state of information processing technology. Because of these dependencies, no specific level of security is implied by the general approval of an algorithm for use in part 1 procedures.

Annex A specifies the assigned values of the Format and Function Code (FFC) to be used when certificates are signed by transforming the text.

Banking — Key management by means of asymmetric algorithms —

Part 2:
Approved algorithms using the RSA cryptosystem

1 Scope

This part of ISO 11166 specifies asymmetric algorithms, i.e. algorithms for asymmetric encipherment and digital signature, employing the RSA cryptosystem and approved for use with ISO 11166-1.

This part of ISO 11166 specifies particular uses of the RSA method. It does not describe the RSA cryptosystem in its general form. Modes of operation for RSA digital signature are also specified. These algorithms and modes of operation are for use only in part 1 of this International Standard.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 11166. 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 11166 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 646:1991, *Information technology — ISO 7-bit coded character set for information interchange.*

ISO 9362:-[1] , *Banking — Banking telecommunication messages — Bank identifier codes.*

ISO 11166-1:1994, *Banking — Key management by means of asymmetric algorithms — Part 1: Principles, protocols and formats.*

3 Definitions and notation

The definitions given in ISO 11166-1, along with the following, apply.

3.1 extended public key : That part of the asymmetric key set which is employed for encipherment and for signature deconstruction.

3.2 extended secret key : That part of the asymmetric key set which is employed for decipherment and for signature construction.

3.3 hash function : A mathematical function operating on a text of indefinite length with a result of fixed length, which has the property that it is computationally unfeasible to produce two texts giving the same result.

3.4 hash result : The result of applying a hash function. In this part of ISO 11166 the hash result is represented by a non-negative integer.

The notation given in ISO 11166-1 for encipherment, decipherment, signature construction and signature deconstruction shall apply. The notation is extended in this part of ISO 11166 to enable a mathematical formulation of these functions to be specified.

4 Symbols and abbreviations

n	Size of modulus, in bits
M	Modulus

NOTE — The abbreviation "mod M" signifies that the result of an arithmetic operation shall be replaced by its least non-negative residue, modulo M.

X	Text for encipherment
Y	Enciphered text
T	Text for signature
h	Hash function
H	Hash result
S	Digital signature
E	Public exponent
D	Secret exponent

[1] To be published (Revision of ISO 9362:1987)

KS1	Extended secret key of the sender of a KSM
KS2	Extended secret key of the receiver of a KSM
KP1	Extended public key of the sender of a KSM
KP2	Extended public key of the receiver of a KSM
P, *Q*	Prime numbers, whose product is *M*
lcm	Lowest common multiple
TR	Text recovery, a signature mode of operation
TH	Text hashing, a signature mode of operation

5 RSA key generation

The values of n, the size of modulus, and E, the public exponent, shall be fixed parameters, to be agreed for each application of RSA key sets.

NOTE — It is recommended that n be a multiple of 8. For rapid encipherment and signature verification, a convenient value is $E = 65\,537$ which rarely restricts the choice of the primes. Where different RSA key sets are used for different purposes (such as encipherment and signature in KSMs and signature in certificates) the key sets for each purpose may have their own parameter values n and E.

Two primes P and Q shall be found, such that their product, which is the modulus M, satisfies the inequalities :

$$2^{n-1} < M = PQ < 2^n$$

The primes P and Q shall be chosen independently by random processes such that there is no number with a greater than 2^{-64} probability of being chosen for P and Q.

Methods of testing and finding primes and of calculating D are not part of this part of ISO 11166. "Probable primes" obtained by primality testing may be used, provided that the probability of choosing a composite number is less than 2^{-30}.

The chosen values of P and Q shall be such that P-1 and Q-1 are each relatively prime to E.

All intermediate values employed in finding primes P and Q and the primes themselves shall be treated as secret values with the same level of security as a secret cryptographic key, in accordance with ISO 11166-1:1994, clauses 1.6 and 1.7.

Secret exponent D shall be the least positive solution of :

$$ED = 1 \bmod \mathrm{lcm}(P\text{-}1, Q\text{-}1)$$

Exponent D shall be treated as a secret value with the same level of security as a secret cryptographic key, in accordance with ISO 11666-1:1994, clauses 1.6 and 1.7.

When $M = PQ$ and secret exponent D have been calculated, all intermediate values shall be destroyed unless a method of computation is to be employed which requires P or Q or values derived therefrom to be retained as secret values. Annex B describes some conditions on P and Q that have been proposed.

The extended secret key comprises the numbers D and M. The extended public key comprises the numbers E and M.

6 RSA encipherment and decipherment

6.1 Encipherment

The text for encipherment shall be expressed as a non-negative integer X which is less than M.

Encipherment shall be the result of the following calculation :

$$Y = X^E \bmod M$$

The result of encipherment is expressed as an integer Y.

6.2 Decipherment

The enciphered text shall be expressed as an integer Y which is less than M.

Decipherment shall be the result of the following calculation :

$$X = Y^D \bmod M$$

The result of decipherment is expressed as an integer X.

7 Modes of operation for RSA signature

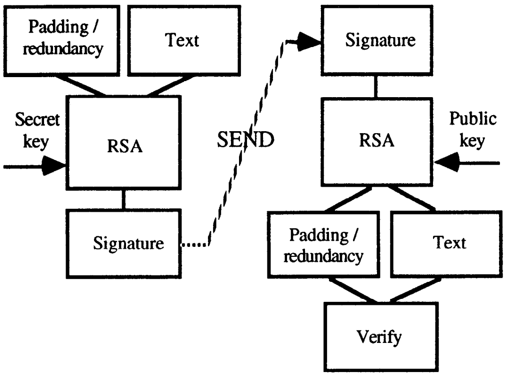
7.1 Text recovery mode of operation (TR)

Figure 1 shows schematically the text recovery mode of operation.

This mode of operation for RSA is an example of signature construction by transforming the text, as specified in ISO 11166-1:1994, subclause 3.3.1.2.(a).

The text recovery mode of operation may only be applied to a block of text represented by an integer T which is less than M.

The method of formation of *T* shall provide redundancy in a form which can be checked during signature deconstruction. See 7.1.2.



a) Contruction b) Deconstruction

Figure 1 — Text recovery mode of signature

7.1.1 Signature construction, TR mode

The digital signature shall be the result of the following calculation :

$$S = T^D \bmod M$$

The resultant digital signature is an integer *S*.

7.1.2 Signature deconstruction, TR mode

Signature deconstruction comprises text recovery and signature verification.

Text recovery shall be the following calculation, which recovers the text *T* :

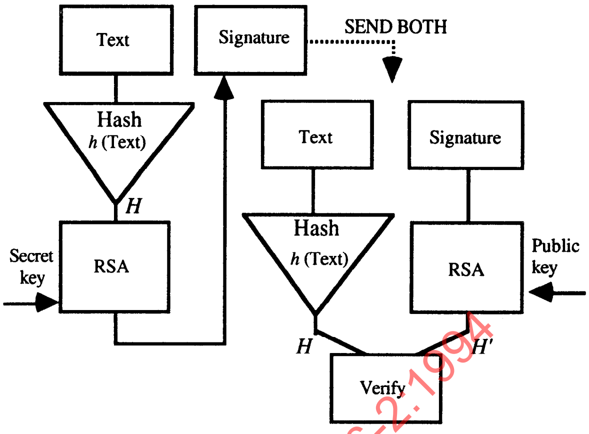
$$T = S^E \bmod M$$

Signature verification shall consist of a check of redundancy in *T* or in data derived from *T*, such that the probability of false verification is less than 2^{-32}.

7.2 Text hashing mode of operation (TH)

Figure 2 shows schematically the text hashing mode of operation.

This mode of operation for RSA is an example of signature construction by means of a separate signature, as specified in ISO 11166-1:1994, subclause 3.3.1.2.(b).



a) Contruction b) Deconstruction

Figure 2 — Text hashing mode of signature

7.2.1 Signature construction, TH mode

The text for signature may be of any size.

An approved hash function *h* shall be applied to the given text *T* to produce the result :

$$H = h(T)$$

The hash function shall be such that the hash result *H* is always less than *M*. The specification of an approved hash function for the purpose of RSA-based signature may include a method of padding the result to match the block size.

The digital signature shall be the result of the following calculation :

$$S = H^D \bmod M$$

The signed text shall comprise both the given text *T* and the signature *S*.

7.2.2 Signature deconstruction, TH mode

The text *T* and signature *S* shall be extracted from the signed text of which they are parts.

Signature verification shall be performed as follows:

A number *H'* shall be calculated as follows :

$$H' = S^E \bmod M$$

Hash function and padding *h*, which were used in signature construction, shall be applied to the text to produce :

$$H = h(T)$$

If the hash function employs an initializing value, this shall be extracted from *H'* and used in the calculation of *h*(*T*).

The signature is valid if $H = H'$.

For signature verification to operate successfully, the recovered text T shall be a bit pattern identical to the text signed by the process in 7.2.1.

8 Application of algorithms

This clause relates the algorithms specified in clauses 6 and 7 to the primitive processes specified in ISO 11166-1:1994, clause 3.3.

Figures 3 and 4 show the way in which these primitives are employed during the generation and processing of a KSM.

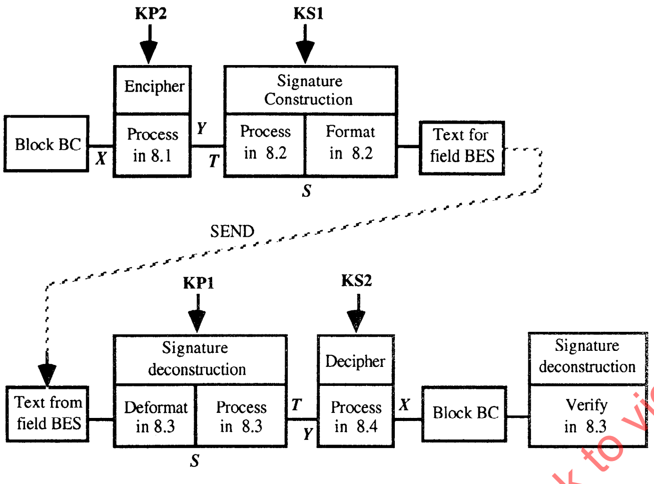


Figure 3 — Cryptographic processing for field BES

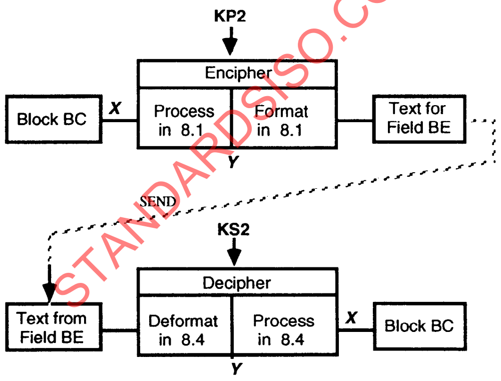


Figure 4 — Cryptographic processing for field BE

8.1 Encipherment of a block

The equation :

enciphered BC = eKP2{BC}

shall identify "BC" with X and "enciphered BC" with Y in the encipherment algorithm specified in 6.1.

KP2 shall be the extended public key of the receiver of the KSM being generated.

Block BC shall be an n bit binary number, constructed by concatenation as specified in ISO 11166-1, with a random fill on the left (most significant bits). The most significant bit shall always be zero, to avoid exceeding the modulus M.

If the resultant enciphered BC is to form the first subfield of BE, it shall be expressed in hexadecimal notation and the hex digits shall be coded by the characters 0,1...F. If the enciphered BC is to be signed as the first subfield of BES, the integer Y shall be the text T for signature.

8.2 Signature construction of an enciphered block

The TR mode of signature shall be used for signature construction of an enciphered block.

The equation :

signed enciphered BC = sKS1{enciphered BC}

shall identify "signed enciphered BC" with S and "enciphered BC" with T in the signature algorithm specified in 7.1.1.

KS1 shall be the extended secret key of the sender of the KSM being generated.

If the value Y resulting from encipherment in subclause 8.1 is greater than or equal to the modulus M of KS1, then BC shall be re-formed with a different random fill and the calculations of 8.1. and 8.2. repeated, until an enciphered block less than M is obtained.

Optionally, by imposing a constraint on the values of RSA moduli, the probability of a repeated calculation being needed may be greatly reduced.

The result S shall be expressed in hexadecimal notation and the hex digits shall be character coded as 0, 1... F to form the first subfield of BES.

8.3 Signature deconstruction of a signed enciphered block

Signature deconstruction shall comprise two processes, text recovery and signature verification.

The characters of the first subfield of BES shall be converted to hexadecimal digits expressed as groups of 4 bits and concatenated to form the value S for signature recovery.

The equation :

enciphered BC = vKP1{signed enciphered BC}

shall identify "enciphered BC" with T and "signed enciphered BC" with S in the text recovery algorithm specified in 7.1.2.

KP1 shall be the extended public key of the sender of the KSM being processed.

Signature verification shall take place firstly by checking the value of CTP in the deciphered block BC and secondly by using the data keys contained in BC to verify the MAC field.

8.4 Decipherment of an enciphered block

When the enciphered block is received as the first subfield of BE, the coded characters shall be converted into hexadecimal digits expressed as groups of 4 bits and concatenated to form Y. When the enciphered block is the result of signature deconstruction applied to the first subfield of BES, this result shall be enciphered text Y.

The equation :

BC = dKS2{enciphered BC}

shall identify "BC" with X and "enciphered BC" with Y in the decipherment algorithm specified in 6.2.

KS2 shall be extended secret key of the receiver of the KSM being processed.

8.5 Signature of the Key Service Message

The TH mode of signature construction specified in 7.2.1. shall be used to produce field SIG of a KSM.

The equation :

SIG = σKS1{text}

shall identify "SIG" with S and "text" with T in the text hashing mode of operation specified in 7.2.1.

KS1 shall be the extended secret key of the sender of the KSM being generated.

The text of the KSM which is used for this signature is specified in table 6 of ISO 11166-1: 1994.

The result S from the process of signature construction specified in 7.2.1. shall be expressed in hexadecimal notation and the hex digits shall be character coded as 0, 1...F to become the field SIG of the KSM.

8.6 Verification of KSM signature

The coded characters of signature field SIG shall be converted to hex digits and concatenated to produce the integer S.

The value of S shall be verified as specified in 7.2.2.

The text of the KSM which is used by the hash function to calculate H is specified in table 11 of ISO 11166-1:1994.

The key KP employed in processing S to calculate H' for signature verification shall be the extended public key of the sender of the KSM being processed.

8.7 Certificate generation and verification

ISO 11166-1 allows the alternatives of signature generation by transforming the text or by a separate signature.

8.7.1 Certificate signature by transforming the text

The certificate text to be transformed depends on the value of the field FFC. The assigned values of FFC for use with RSA signature are specified in annex A , together with the corresponding certificate text formats.

The signature construction and deconstruction procedures are specified in ISO 11166-1, subclause 3.3.4.1, with the following interpretation of the primitive processes.

The equation :

certificate signature = sKSC{certificate text}

shall identify "certificate text" with T and "certificate signature" with S in the signature algorithm specified in 7.1.1.

KSC shall be the extended secret key of the CKC.

The equation :

certificate text = vKPC{certificate signature}

shall identify "certificate text" with T and "certificate signature" with S in the signature deconstruction algorithm specified in 7.1.2.

KPC shall be the extended public key of the CKC.

The signature is valid if the recovered text meets the requirements specified in annex A .

8.7.2 Certificate with a separate signature

The equation :

certificate signature = σKSC{certificate text}

is applied to the certificate text specified in ISO 11166-1:1994, subclause 3.3.4.3, to produce the signature which is the first subfield of a CV field.

"Certificate text" shall be identified with *T* and the resultant signature with *S* in the signature construction and deconstruction processes specified in 7.2.1. and 7.2.2.

The extended secret key of the CKC shall be used for certificate signature construction and the extended public key of the CKC for certificate signature deconstruction.

Annex A
(normative)

Assigned values of Function and Format Code (FFC)

A.1 Scope

This annex defines the certificate text and processing specific to certificate signatures formed by transforming the text in accordance with subclause 8.7.1. and ISO 11166-1:1994, subclause 3.3.4.

A.2 Architecture

The certificate text shall contain the owner's Public Key (the modulus, *M*, from the Extended Public Key) preceded by identifying data defined by this annex. The first octet of the identifying data is a Format and Function code which defines the content and format of the certificate text together with the function of the certified extended public key as specified in ISO 11166-1:1994, subclause 3.3.4.

The total length (in bits) of the identifying data (including FFC) is equal to the difference in length between the CKC Public Key used to deconstruct the certificate signature and the Public Key contained in the certificate text. Table A.1. gives the length of the identifying data for each format of certificate text described in this part of the standard. This length shall always be a multiple of 64 bits.

A.3 Character set and representation

Fields within the certificate text shall be constructed from characters of one of the types defined in this clause. All fields within certificate text are of fixed length defined by the FFC.

A.3.1 Alphanumeric characters

Alphanumeric characters are formed by taking the least significant six bits of the corresponding character in ISO/IEC 646. The character set for all alphanumeric fields shall comprise the character 0-9, A-Z and space (b). Fields containing alphanumeric characters shall be a multiple of four characters (3 octets) long and shall be padded (if necessary) with trailing spaces. This shall be the one use of the space character.

A.3.2 Numeric characters

Numeric characters shall be formed using 4 binary bits, the leftmost bit being the most significant. The character set for numeric fields shall be either hexadecimal or decimal. Hexadecimal characters shall be represented by symbols 0-9 and A-F. Decimal characters shall be represented by symbols 0-9. Numeric fields shall be a multiple of two characters (1 octet) and shall be padded if necessary with leading binary 0s.

A.3.3 Binary fields

Fields containing binary values shall be a whole number of octets long and shall be padded if necessary with leading binary 0s.

A.4 Verification of fields

As part of the process of verifying a certificate signature, the format of fields extracted from the certificate text and of a type identified in this clause shall be verified using the following rules.

A.4.1 Alphanumeric fields

A certificate signature shall be rejected as invalid if any field specified as alphanumeric contains a character which is not one of these specified in A.3.1.

A.4.2 Date and time fields

A certificate signature shall be rejected as invalid if any field specified as containing a time and/or date contains a value other than 0 to 9. In addition, the component parts of the date/time shall be within the following ranges :

Year (YY)	:	00 to 99
Month (MM)	:	01 to 12
Day (DD)	:	01 to 31
Hours (HH)	:	00 to 23
Minutes (MM)	:	00 to 59
Seconds (SS)	:	00 to 59

NOTE — When checking that a certificate text contains the correct redundancy, it is sufficient to check that the day has a value of 01 to 31. However before using the recovered day, it may be necessary to interpret the date and reject it if the day value is inconsistent with the month or year.

A.4.3 Other decimal numeric fields

A certificate signature shall be rejected as invalid if any field specified as containing decimal digits contains a value other than 0-9.

A.4.4 Fixed value field

A certificate signature shall be rejected as invalid if any field specified as containing a fixed value contains a value other than that specified.

A.5 Processing of certificate text

The certificate text shall be recovered from the certificate signature using the deconstruction process specified in 8.7.1. The FFC is the first octet of the recovered certificate text. The least significant two bits define the function of the Public Key in the certificate text (FKP) as defined in table 2 of ISO 11166-1:1994. The remaining bits define the format of the text. Table A.1. provides an index to the table containing the definition for the specific certificate format. Information in the relevant table shall be used to extract the fields and verify the format of the fields as defined in clause A.4. Finally the checks specified in ISO 11166-1:1994, subclause 3.3.4.5, shall be performed.

Table A.1 — Index to FFC

FFC (hex value)	Difference in Modulus size (bits)	Prime characteristics	See table
00 to 03	128	Owner name is an 8-character Bank Identifier Code registered in accordance with ISO 9362	A.2
10 to 13	64	Minimal length certificate signature	A.3
20 to 23	256	Long certificate including effective from and expiry dates	A.4

Table A.2 — Certificate Text for FFC = 00

Field	Name	Specification
FFC	Format and Function Code	Two hexadecimal digits
	Fixed value (reserved for future use)	Two hexadecimal digits of value 0
EXCV	Expiry date of certificate	4 decimal digits in the order YYMM
OWN	The name of the party to whom the certificate was originally issued	Eight alphanumeric characters specifying a Bank Identifier Code registered in accordance with ISO 9362
-	Fixed padding	8 alphanumeric space (b̲) characters
PK	Public Key	The binary value of the modulus, M, of the owner's Extended Public Key

Table A.3 — Short certificate

Field	Name	Specification
FFC	Format and Function Code	Two hexadecimal digits
EXCV	Expiry date of certificate	4 decimal digits in the order YYMM
OWN	The name of the party to whom the certificate was originally issued	4 decimal digits
-	Fixed padding	6 digits of value 0
PK	Public Key	The binary value of the modulus, M, of the owner's Extended Public Key

Table A.4 — Long certificate

Field	Name	Specification
FFC	Format and Function Code	Two hexadecimal digits (1 octet)
	Fixed value (reserved for future use)	Two hexadecimal digits of value 0 (1 octet)
EDCV	Effective date and time of certificate	12 decimal digits (6 octets) in the order YYMMDDHHMMSS
EXCV	Expiry date and time of certificate	12 decimal digits (6 octets) in the order YYMMDDHHMMSS
OWN	The name of the party to whom the certificate was originally issued	16 alphanumeric characters (12 octets)
IDCV	The identity of the certificate	A 12 decimal digit number issued by the CKC to identify the certificate (6 octets)
PK	Public Key	The binary value of the modulus, M, of the owner's Extended Public Key

Annex B
(informative)

Choice of primes

B.1 Conditions for the choice of primes

There are special conditions under which factorization of the product $M = PQ$ is easy, for example if P and Q differ by a small number. This, and other similar adverse conditions have a very low probability of occurrence if the choice of P and Q is uniformly distributed over the available range.

Conditions on the choice of P and Q have been proposed, to avoid these problems without relying on their low probability of occurrence, such as:

- P and Q should differ by a large number.
- P-1 and P+1 should each have a large prime factor.
- If R is the large prime factor of P-1, then R-1 should have a larger prime factor.
- Q also should satisfy the conditions stated for P.

B.2 Uniformly distributed primes

There is a requirement in clause 5 that the primes P and Q shall be chosen independently by random processes such that there is no number with greater than 2^{-64} probability of being chosen for P and Q. This condition ensures a sufficiently large key space for the secret key.

One method of achieving this result is to choose each prime at random from all the primes in a certain range. In order to meet the conditions given in clause B.1, it may be necessary to restrict the primes to those in a particular arithmetic progression (see Gordon, J., 'Strong RSA keys', Electronics Letters, 20 June 1984, 514-516), but the same principles apply.

This annex describes a method for such 'uniformly distributed' primes as an example of one way to meet the above mentioned requirement in clause 5.

This part of ISO 11166 requires that the modulus lies in a certain range, and this can be achieved by choosing suitable ranges for the two primes, P and Q. The modulus is to be of size n bits with the most significant bit a 1, that is:

$$2^{n-1} < M < 2^n$$

This can be achieved, for example, by setting these ranges for P and Q:

$$3 \times 2^{x-2} < P < 2^x, \quad 3 \times 2^{y-2} < Q < 2^y,$$
where $x+y = n$.

The range of M has been reduced from the optimum, but this is of little consequence. By choosing different numbers for x and y, the condition that P and Q shall differ by a large number is met, and it is best that x and y should differ by the smallest amount, either 1 or 2.

Having made a choice of ranges for P and Q, the method of choosing a prime uniformly distributed among the primes in a certain range will be described for the case of prime P.

A random variable is required, which may be obtained in a number of ways, for example from a true random source such as can be generated from electrical noise. Another popular method is to time very accurately an operator's key depressions on a keyboard and use the timings in a pseudo-random process which generates the required random variable. The method of production must be examined to ensure that its uncertainty (entropy, when regarded as a source in information theory) is sufficiently high. The random variable must be handled with the same precautions as for a secret key.

Given that prime P is to be chosen within a fixed range $a < P < b$, the example method proposed here consists of testing candidate numbers in this range for primality until a prime is found. For simplicity, the case considered here is that all odd numbers are candidates.

A random number r with uniform distribution is required to start the process. Let this be chosen randomly among the integers:

0, 1, 2,... (b/2-a/2-1)

then $c_1 = a+1+2r$ is a randomly chosen odd number in the range between a and b (both a and b are even numbers). Let c_0 be the first candidate number, which is a+1 in this case.

1) set $c = c_1$

2) test c for primality

   if prime, set $P = c$, and the task is complete

   if not, increment c by 2 to find the next candidate.

3) if $c > b$, set $c = c_0$

4) if $c = c_1$, the search has failed.

5) return to step 2.

Steps 3 and 4 are relevant to the search among a restricted set of primes to meet the conditions given in clause B.1, where the calculations of c_0 and c_1 are different from those given above and the increment is a large number. The reference quoted shows how the probability of failure (requiring a repeat of earlier phases of the process) can be kept small.

Annex C
(informative)

Bibliography

ANSI X3.92:1981, *American National Standard for Information Systems — Data Encryption Algorithm.*

Rivest, R.L., Shamir, A. and Adleman, L., *A method for obtaining digital signatures and public key cryptosystems — Communications of the ACM,* 21, 2 (February 1978), 120-126.