Title: PROTECTING MODULAR EXPONENTIATION IN CRYPTOGRAPHIC OPERATIONS

Abstract: The present invention proposes a method for executing a blinded modular exponentiation, based on a window method with a window size of k bits so using $2^h$ pre-calculated variables ($Y_i = X_i^h \mod N$ for $i = 0 \to 2^{h-1}$), on input data X of n bits to obtain output data S of n bits, $S = X^d \mod N$, where d is the exponent of size m bits and N is the modulus of n bits, comprising the steps of: - blinding the pre-calculated variables by a blinding value $B_i$ being a pseudo-random variable of the size of the modulus (n bits) and lower than the modulus ($y_j = Y_j \times B_i \mod N$ for $i = 0 \to 2^{h-1}$) - executing the modular exponentiation with the blinded pre-calculated variables, to obtain an intermediate result (A), - unblinding the intermediate result by a unblinding value $C_i = (B_i)^{-1} \mod N$ where g equals the concatenation of m/k times the value "$1$" coded on k bits, to obtain the output data $S$. 

**Fig. 1**

---

(51) International Patent Classification: G06F 7/72 (2006.01)

(21) International Application Number: PCT/EP2011/066952

(22) International Filing Date: 29 September 2011 (29.09.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
- 10290532.0 4 October 2010 (04.10.2010) EP


(72) Inventor: and

(74) Agent: LEMAN CONSULTING S.A. 284; Chemin de Precoxy 51, CH-1260 Nyon (CH).


Published: with international search report (Art. 21(3))
PROTECTING MODULAR EXPONENTIATION IN CRYPTOGRAPHIC OPERATIONS

FIELD OF THE INVENTION

The invention relates to software and data cryptography. In particular, the invention relates to a method for hiding intermediate results of a modular exponentiation.

INTRODUCTION

Till not so long ago, cryptography was concerned only by the protection of the communication of the message into a hostile environment. In classical scheme (a.k.a. black-box model), the attacker had only access to the inputs of the decryption device. With the emergence of Pay-TV, digital contents protected by DRM (movie, music in smart-phone, personal computer or in CD/DVD), attacker has now physically access to the decryption device and its outputs meaning that not only he can passively study the state and intermediate values of the encryption device, but also actively affect its computations.

Specifically, in 1996 appeared the notion of fault analysis: when submitting the decryption device to abnormal conditions (wrong input, abnormal temperature, strong electromagnetic radiations... ), the decryption algorithm can output faulty plaintext which gives information about the key used in the decryption device. See "On the Importance of Checking Cryptographic Protocols for Faults" of Dan Boneh, Richard A. DeMillo and Richard J. Lipton in the proceedings of Eurocrypt 1997.

In the same year appeared the notion of side-channel attacks: the physical signals (timing of processing, power consumption, electromagnetic radiations... ) that are output by the decryption device during the processing of the decryption can leak information (side-channel information) about the internal variables of the decryption algorithm. From this internal variables and statistic analysis, the attacker can retrieve information about the key used in the decryption device. See "Timing Attacks on Implementations of Diffie-
Hellman, RSA, DSS, and Other Systems" of Paul C. Kocher in the proceedings of Crypto 1996.

Fault analysis and side-channel analysis belong to the grey-box model: the attacker has a limited knowledge about the implementation of the cryptographic algorithm and about its internal data. These attacks were successfully used to retrieve the keys and code source of smartcards used in pay-TV systems.

For music and movie on personal computer or on CD/DVD, the content keys are protected by obfuscation of software (DRM) because it is much less expensive than to distribute smartcards. In this case the environment is even more hostile than in the grey-box model, the attacker has a full access of the inner part of the software. This is what is called the white-box model. In 2002 appeared the concept of White-Box Cryptography. White-Box Cryptography is an obfuscation technique intended to implement cryptographic primitives in such a way, that even an adversary who has full access to the implementation and its execution platform, is unable to extract key information [1].

As described in the thesis of Brecht Wyseur about White-Box Cryptography, a countermeasure that is efficient against attacks in the white-box model is also efficient against attacks in the grey-box model.

PRIOR ART

Modular exponentiation is involved in some important cryptographic protocols for key exchange or encryption or signature (Diffie-Hellman, ElGamal, RSA, DSS...). It is well known in the art that the most basic method to perform a modular exponentiation is the so-called "square-and-multiply" algorithm [2] which consists in processing the exponent bit by bit and performing multiplication according to its value. In the following we review some of the notions relative to the notion of modular exponentiation as well as some of the state-of-the-art algorithms.

- **Exponentiation** is a mathematical operation, written as $b^e$, involving two numbers, the **base** $a$ and the **exponent** $e$. When $e$ is a **positive integer**, exponentiation corresponds to repeated multiplication; in other words, a product of $e$ factors of $b$: $b \times b \ldots$
• **Modular exponentiation** is a type of exponentiation performed over a modulus. Doing a "modular exponentiation" means calculating the remainder when dividing by a positive integer m (called the modulus) a positive integer b (called the base) raised to the e-th power (e is called the exponent). In other words, problems take the form where given base b, exponent e, and modulus m, one wishes to calculate c such that: \(c = b^e \mod m\)

• **Basic binary modular exponentiation (square-and-multiply)**

INPUT: an integer g of n bits and integer e of \((t+1)\) bits i.e. \(e = (e_t e_{t-1} ... e_1 e_0)\)
where \(e_i\) is the i-th least significant bit of e, m the modulus of n bits.

OUTPUT: \(g^e \mod m\)

1. \(A = 1\).

2. For i from t down to 0 do the following:
   1.1 \(A = A \times A \mod m\).
   2.1 If \(\beta_i = 1\), then \(A = A \times g \mod m\)

3. Return(A)

The binary exponentiation can be substantially speed up by analysing the exponent by group of k bits. This method is also known as the **window method**, where the window size is k.

• **Modular exponentiation with window method**

INPUT: g of n bits, e of \((t+1)^*b\) bits i.e \(e = (e_t e_{t-1} ... e_1 e_0)b\), where \(b = 2^k\) for some \(k > 1\) and a modulus m of n bits.

OUTPUT: \(g^e \mod m\)

1. **Precomputation.**
   1.1 \(G_0 = 1\).
   1.2 For i from 1 to \((2^k - 1)\) do: \(G_i = G_{i-1} \times g\). (Thus, \(G_i = g^i\).)

2. \(A = 1\).

3. For i from t down to 0 do the following:
\[ 3.1 \quad B = A^2 \mod m, \quad A = B^k \mod m \]

\[ 3.2 \quad j = e_i, \quad A = A \times G^j. \]

4. Return(A)

It should be emphasized that the goal of the attacker in any of the above models is to obtain the secret key in order to use it for its own illegal purposes. For modular exponentiation, if the exponent is properly protected by obfuscation techniques, the attacker will try to have information about the key by monitoring the intermediate results of the modular exponentiation. Those skilled in the art would notice in fact that in the case of the basic square-and-multiply method the secret key value (the exponent) can be trivially obtained by the attacker by simply observing the execution of the exponentiation algorithm and measuring the time (or the power consumed) of every step involved in the computation. The window method is also prone to some advanced side channel attack techniques such as Differential Power Analysis (DPA) described in [3].

Prior art for obfuscating cryptographic computations include a method proposed in [4]. The advantage of the present invention compared to the solution presented in [4] is that it requires no additional computations in the main loop except the "blinding" of the precomputed window factors hence performing the window-based exponentiation substantially faster.

While some number of other method for obfuscating and securing modular exponentiation operation were proposed in the prior art such as, for instance [5], they all do perform a masking (blinding) of the encrypted message itself (C) or the decryption exponent (cf). Present invention proposes a new method where the masking is applied on the pre-calculated window values making it much more difficult to the attacker to bypass the blinding by one fault attack or software modification. Those skilled in the art understand that by "blinding" or "masking" the operation of randomization of a variable or a value is assumed such that the said variable or value frequently changes a hence cannot be identified and studied by an attacker using side channel attack methods.

PROBLEM TO BE SOLVED
The aim of the present invention is to provide a way to blind the intermediate results of the modular exponentiation in such a way that the blinding method is more difficult to bypass by advanced side channel analysis that the blinding method described in known art such as [4] and [5] as well as in "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems" of Paul C. Kocher in the proceedings of Crypto 1996.

BRIEF DESCRIPTION OF THE INVENTION

The present invention proposes a method for protecting modular exponentiation, based on a window method with a window size of k bits so using $2^k$ pre-calculated variables ($Y_i = X^i \mod N$ for $i = 0$ to $2^{k-1}$), on input data X of n bits to obtain output data S of n bits, $S = X^d \mod N$, where d is the exponent of size m bits and N is the modulus of n bits, comprising the steps of:

- blinding the pre-calculated variables by a blinding value $B_i$ being a pseudo-random variable of the size of the modulus (n bits) and lower than the modulus ($Y_j = Y_i \times B_i \mod N$ for $i = 0$ to $2^{k-1}$)
- executing the modular exponentiation with the blinded pre-calculated variables, to obtain an intermediate result (A),
- unblinding the intermediate result by a unblinding value $C_i = (B_i^{-1})^{\mod N}$ where g equals the concatenation of m/k times the value "1" coded on k bits, to obtain the output data S.

BRIEF DESCRIPTION OF THE FIGURE

The present invention will be better understood thanks to the attached figure showing a processing unit able to execute the various steps of the claimed method.

DETAILED DESCRIPTION OF THE INVENTION AND PREFERRED EMBODIMENT

The present invention describes a method for protection for a modular exponentiation operation using the so-called window method in an open software environment. By an open software environment we assume binary
code which is executed on the said PC system and which can be accessed by an attacker.

This invention can be implemented in a processing unit dedicated to execute cryptographic operations as illustrated in the figure 1. This unit comprises at least a processor CPU able to execute a software core and a memory MEM1 to store this code and provide the space necessary to store the temporary data MEM2. An interface INT is provided so as to receive the messages encrypted (or decrypted) to be stored in the temporary memory MEM2 for crypto processing. In the same manner, the interface INT can transmit the messages decrypted (or encrypted) to the other components of the reception device.

According to the preferred embodiment we consider a PC system or a processing unit which executes the said modular exponentiation operation using window method the said method implemented in the said software environment. Let X be the input data of n bits and K be the key which comprises an exponent d having m bits and a modulus N having n bits. The modular exponentiation operation implemented in the said PC system comprises two steps: pre-calculation and exponentiation. During the pre-calculation step values $Y_i = X^i \mod N$ are pre-computed. Those skilled in the art notice that in the window method i varies between 0 to $2^{k-1}$ and k represent the size of a window applied to the exponent d. During the exponentiation step the said pre-computed values $Y_i$ are used.

Below the implementation of the invention is described in pseudocode. It is important to note that the steps described below are solely presented for the purpose of the preferred embodiment of the present invention and are not, in any case, limiting.

Use a register A of n bits, initialized with the value 1, for temporary storage of intermediate results of the exponentiation algorithm.

Use a register C of n bits, initialized with the value 1, for temporary storage of the last used random value.

Use a register v of $\log(m)$ bits, initialized with the value $m/k$
- for j = m/k to 1, extracting k bits of the key d starting by the bit jxk,

Determine a blinding condition (0/1) b based on a function f1 of j,

If the blinding condition is set, execute:

- update a pseudorandom variable B of n bits
- for i = 0 to 2^k-1, replacing the Yj by Y × (B mod N),
- for i = 0 to 2^k-1, replacing Y by Y × (C^{-1} mod N),
- Replace A by A × C^g in which g is a function of (v-j) (i.e. g = f2(v-j), where f2(r) is a number of k × r bits build from the concatenation of r times the number 1 coded on k bits e.g. if r = 4 and k = 2, then f2(r=4) = "01 0 10 10 1" in binary, f2(4) = 0x55 in hexadecimal)
- Load the register C with the value of B,
- Load the register v with the value of j

Replace the value A by (A^K mod N * Y_{dj} mod N), dj representing the j^{th} word of k bits of the exponent d when counting from the least significant bit to the most significant bit and K = 2^k

The person skilled in the art would apprehend the advantage of this blinding method which consists in the fact that the blinding is involved in several computations (as many computations as the size of the windows), so it is more difficult for a side channel attacker to bypass the blinding by one fault attack or one software modification.

Another advantage with respect to the known prior art is the renewability of the blinding inside the algorithm which makes it more difficult to bypass the blinding for a fault attack or by means of software modifications.

The method according to claim 1, where the blinding value Bi is renewed after the processing of several windows and the unblinding of the intermediate result is done by the multiplication by a variable CI which depends of the size of the window (k), the number of windows which were processed (w), the modulus N and the initial blinding value Bi: CI = (Bi^h)^t mod N where h equals the concatenation of w times the value "1" coded on k bits.
Another advantage is that a hacker can not find a specific function \( T \) such that submitting \( T(X) \) as input of the modular exponentiation will make the intermediate results stored in \( A \) independent of the blinding variable \( B \), even if \( B \) is a known constant.

The overhead of the countermeasure if the unblinding steps are not pre-computed, compared to the classical exponentiation algorithm, is an exponentiation with the same length than the input exponent \( d \) and as many inversions as the number of blinding updates.

According to another particular embodiment, when implemented on the said PC system the claimed method can be simplified when constraints on speed exist by blinding the pre-calculated variables once at the very beginning of the exponentiation algorithm and then removing the blinding at the end of the exponentiation. The overhead of the countermeasure is then one exponentiation with the same length than the input exponent \( d \) and only one inversion.

Those skilled in the art know that the modular exponentiation method is usually used in the context of RSA cryptosystem. According to a particular embodiment, when the modular exponentiation algorithm is used for RSA computation with a private key \( d \) (which might be relatively large in size in terms of bits), the claimed method can be speed up according to the followings steps:

- pre-computing \( e^i = g^{i} \mod (p-1) \ast (q-1) \) where \( g \) equals the concatenation of \( m/k \) times the value "1" coded on \( k \) bits.

- blinding of the pre-calculated variables \( (Y_i = X^i \mod N) \) by a same blinding value \( B_2 \) such that \( B_2 = B_i^e \mod N \), \( B_i \) being a pseudo-random variable of the size of the modulus and lower than the modulus \( (Y_i = Y_i \times B_2 \mod N \text{ for } i = 0 \text{ to } 2^k - 1) \)

- executing the modular exponentiation with the blinded pre-calculated variables, to obtain an intermediate result \( A \),

- unblinding of the intermediate result by the inverse of \( B_i \)
By this way, the blinding overhead is reduced to one exponentiation with a
small exponent (exponentiation by e') and one inversion.

According to the preferred embodiment the pseudorandom variable B can be
renewed by different methods. Furthermore values of B used between two
executions of the main method can also be different. Below such
implementations are described in accordance with the preferred embodiment.

One way of speed up is to create a link between the new and the previous
blinding. In the case of the blinding value B is renewed after the processing of
w windows, the blinding values used during the exponentiation is an array of
sub-blocks \( B = (b_1, b_2, b_3, \ldots, b_n) \), the subsequence sub-block \( B_{i+1} \) being the
square value modulo \( N \) of the preceding \( b_i \), each sub-block \( b_i \) being a
pseudo-random variable of the size of the modulus and lower than the
modulus, the unblinding values used during the exponentiation is an array of
sub-blocks \( C = (c_1, c_2, c_3, \ldots, c_n) \), the subsequence sub-block \( C_{i+1} \) being the
square value of the preceding \( c_i \). \( C_i = (B, g)^{-1} \mod N \) where \( g \) equals the
concatenation of \( w \) times the value “1” coded on \( k \) bits but only \( C_i \) is
computed using the inversion, the other \( C_i \) being the square of the preceding.

\( B \) can be updated inside the main algorithm but the same values for \( B \) can be
used between two executions of the main method: the different values of \( B \)
and of \( C^9 \) can thus be pre-computed to remove the overhead of the
exponentiation and of the inversions.

In a particular case of the preferred embodiment, the method comprises a
step of pre-computing and storing the value \( C^9 \) where the blinding value \( B \) is a
digest of all or part of the modular exponentiation code.

REFERENCES

[1] "White-Box Cryptography and an AES Implementation" of Stanley Chow,
Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot in the proceedings
of the 9th International Workshop on Selected Areas in Cryptography (SAC
2002)

Vanstone.


CLAIMS

1. A method for protecting modular exponentiation in cryptographic operations executed by a processing unit, said modular exponentiations being based on a window method with a window size of k bits, using $2^k$ pre-calculated variables ($Y_i = X^i \mod N$ for $i = 0$ to $2^k-1$), on input data X of n bits to obtain output data S of n bits, $S = X^d \mod N$, where d is the exponent of size m bits and N is the modulus of n bits, comprising the steps of:

   • blinding the pre-calculated variables by a blinding value $B_i$ being a pseudo-random variable of the size of the modulus (n bits) and lower than the modulus ($Y_j = Y \times B_i \mod N$ for $i = 0$ to $2^k-1$)
   
   • executing the modular exponentiation according to the window method which is based on the division of the exponent d into blocks of size of at most k bits representing a window, with the said blinded pre-calculated variables to obtain an intermediate result (A),
   
   • unblinding the intermediate result by a unblinding value $C_i = (B_i^g)^{-1} \mod N$ where g equals the concatenation of $m/k$ times the value "1" coded on k bits, to obtain the output data S.

2. The method according to claim 1, where the window method is based on the division of the exponent d into blocks of at most k bits representing a window, and the blinding value $B_i$ is renewed after the processing of one or more said blocks and the unblinding of the intermediate result is done by the multiplication by a variable $C_i$ which depends of the size of the window (k), the number of windows which were processed (w), the modulus N and the initial blinding value $B_i$: $C_i = (B_i^h)^{-1} \mod N$ where h equals the concatenation of w times the value "1" coded on k bits.

3. The method according to claim 1, where the modulus N is the product of two primes p,q of n/2 bits, comprising the steps of:

   • pre-computing $e^* = g^{-1} \mod (p-1) \times (q-1)$ where g equals the concatenation of $m/k$ times the value "1" coded on k bits.

   • blinding of the pre-calculated variables ($Y = X \mod N$) by a same blinding value $B_2$ such that $B_2 = B_i^e \mod N$, $B_i$ being a pseudo-random
variable of the size of the modulus and lower than the modulus ($Y_i = Y_i \times B_2 \mod N$ for $i = 0$ to $2^k - 1$)

- executing the modular exponentiation with the blinded pre-calculated variables, to obtain an intermediate result ($A$),

- unblinding of the intermediate result by the inverse of $B_2$

4. The method according to claim 1 or 2 or 3, where the blinding value $B_i$ is a dynamic random value which is updated at each execution of the steps of the modular exponentiation with the blinded pre-calculated variables.

5. The method according to claim 2, where the blinding value is renewed after each processing of window of $k$ bits, the blinding values used during the exponentiation is an array of sub-blocks $B = (B_1, B_2, B_3, \ldots, B_n)$, the subsequent sub-block $B_{i+1}$ being the square value modulo $N$ of the preceding $B_i$, each sub-block $B_j$ being a pseudo-random variable of the size of the modulus and lower than the modulus, the unblinding values used during the exponentiation is an array of sub-blocks $C = (C_1, C_2, C_3, \ldots, C_n)$ the subsequent sub-block $C_{i+1}$ being the square value of the preceding $C_i$, $C_i = (B_i^h - 1) \mod N$ where $h$ equals the concatenation of $w$ times the value "1" coded on $k$ bits but only $C_i$ is computed using the inversion, the other $C_i$ being the square of the preceding.

6. The method according to any one of the previous claims, where the blinding value $B_i$ is a static pseudorandom value and the unblinding value is pre-computed according to this static value once for multiple execution of the method of claim 1 or 3 or 5.

7. The method according to any one of the previous claims, comprising the step of precomputing and storing the value $C_i$ where the blinding value $B_i$ is a digest of all or part of the modular exponentiation code.
INTERNATIONAL SEARCH REPORT

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. □ Claims Nos.: because they relate to subject matter not required to be searched by this Authority, namely:

2. X claims Nos.: 2, 3, 6, 7 (completely); 5 (partially)
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

   see FURTHER INFORMATION sheet PCT/ISA/210

3. □ Claims Nos.: because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. □ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. □ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of additional fees.

3. □ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. □ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

□ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

□ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

□ No protest accompanied the payment of additional search fees.
INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/066952

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F7/72
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>page 11, line 29 - page 12, line 6; figures 2, 5</td>
<td>5</td>
</tr>
</tbody>
</table>

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :
   *A* document defining the general state of the art which is not considered to be of particular relevance
   *E* earlier document but published on or after the international filing date
   *L* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
   *G* document referred to in oral disclosure, use, exhibition or other means
   *P* document published prior to the international filing date but later than the priority date claimed
   *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
   *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
   *A* document member of the same patent family

Date of the actual completion of the international search: 22 December 2011

Date of mailing of the international search report: 05/01/2012

Name and mailing address of the ISA:
European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel: (+31-70) 340-2040, Fax: (+31-70) 340-3016

Authorized officer: Verhoof, Paul

Form PCT/ISA/210 (second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>P. KOCHER: &quot;TIMING ATTACKS ON IMPLEMENTATIONS OF DIFFIE-HELLMAN, RSA, DSS, AND OTHER SYSTEMS&quot;, ADVANCES IN CRYPTOLOGY - CRYPTO '96; LNCS, vol. 1109, 18 August 1996 (1996-08-18), pages 104-113, XP000626590, Springer Verlag, Berlin, Germany DOI: 10.1007/3-540-68697-5</td>
<td>1, 5</td>
</tr>
<tr>
<td>A</td>
<td>US 2003/133567 AI (YAJIMA JUN [JP] ET AL) 17 July 2003 (2003-07-17), paragraph [0124]; figure 3</td>
<td>1</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2007006649 Al</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2003208097 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2003133567 Al</td>
</tr>
</tbody>
</table>
Continuatio n of Box II.2

Claims Nos. : 2, 3, 6, 7 (completely) ; 5 (partially)

The inventi on as claimed accordi ng to claims 2, 3, 6 and 7 (entirely) and claim 5 (partially) is not di sclos ed in a manner suffi ciently clear and complete for i t to be carri ed out by a person skille d in the art, contrary to Arti cle 5 PCT. Furthermore, these claims are not fu lly supported by the descri ption, as requi red by Arti cle 6 PCT. These lack of disclosure and lack of support are such that no meaningful search is possible for claims 2, 3, 6 and 7 and that claim 5 can only be partially searched. The reasons therefore are the following: The descri ption on does not comply with Rule 5.1 PCT in that the part relating to Rule 5.1(v) is not present. The descri ption on does not go beyond the claims, apart from stating some probl ems solved (Rule 5.1(iii) PCT). No examples are given, which makes it impossible to fall back on the descri ption, where the claims are not readil y comprehensible. Claims 2, 6 and 7 The subject-matter of claims 2, 6 and 7 is complet ely absent from the descri ption. Furthermore, these claims are not readil y comprehensible, for the following reasons: Claim 2 states that the value of C 1 depends on w. Yet, claim 2 depends on claim 1, which gives a formula for Cl which does not have w in it. The value for C given in claim 2 is different from the one in claim 1, thereby contradicting it. Moreover, in the absence of any example, nor any techni cal advantage mentioned in the description, the skilled person would not be able to implement the method of claim with the formul a for Cl given in claim 2 instead of the one in claim 1. This is because claim 2 does not say how the binding value B is renewed. This cannot be remedied by amendment, hence even a partial search would not be meaningful. In claim 6, the term "statis c pseudorandom value" is not comprehensible. If this value does not change, it cannot be a pseudorandom value, as the term "pseudorandom" relates to a sequence of random values. Furthermore, this claim refers both to "any one of the previous claims" and to "claim 1 or 3 or 5", such that the claim dependency is unclear. In the absence of any example, nor any technical advantage mentioned in the descri ption, the skilled person would not be able to implement the method of claim 6. This cannot be remedied by amendment, hence even a partial search would not be meaningful. In claim 7, the terms "digest" and "exponentiation code" are not readily comprehensible. In the absence of any example, nor any technical advantage mentioned in the description, the skilled person would not be able to implement the method of claim 7. This cannot be remedied by amendment, hence even a partial search would not be meaningful. Claim 3 is unclear (Arti cle 6 PCT) in that it is incompatible with claim 1, despite the fact that it is dependent on it. For the analysis whether a meaningful search can be done, claim 3 is interpreted as princi pal part of the features of claim 1, hence being an independent claim. Furthermore, in the light of the descri ption on, p. 8, 1. 30, the last line of claim 3 is understood to involve B 1 instead of B 2 as otherwi se the implementation on cannot be understood to work: (B 2 )-l ? (B 2 )? 1. The technical probl em addressed by claim 3 is not clear from the application as a whole, contrary to Rule 5.1(iii) PCT. For coding the value e in bits are needed, in view of the fact that (p-1) ?(q-1) needs n bits. In RSA p and q are big prime values. The calculation of B 2 is thus very resource
intensive. Also, the calculation of the inverse of Bl and of g1s of the same order. Yet, the technical advantage given on page 9, 1, 1-2 is that "the blinding overhead is reduced to one exponentiation with a small exponent (exponent at on by e") and one inversion". This cannot be understood. Consequently, it is not clear from the application as a whole how this technical advantage of the invention as claimed can be achieved, contrary to Article 17(2) PCT. In the absence of any technical effect, the additional features of claim 3 do not have technical character, such that a meaningful search is not possible. Claim 5 In the light of the description, p. 9, 1, 9, the second line of claim 5 is understood to involve windows as otherwise the implementation would be a complete waste of resources, not possibly qualifying as inventive. Nevertheless, the bottom three lines of claim 5 do not seem to result in a workable implementation, unless w=m/k (it would seem that if w<m/k, the w+l-th and subsequent blocks of k bits would have to have the value 2 coded on k bits.) In the absence of any example, nor any technical advantage mentioned in the description, the skilled person would not be able to implement the method of claim 5 for the whole ambit of the claim. This claim can only be searched with w being an integer multiple of m/k, such that h=g.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on a matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guidelines C-VI, 8.2), should the problems which led to the Article 17(2) declaration be overcome.