(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 832 007**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **01 14272**

(51) Int Cl[7] : **H 04 L 9/18**

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 05.11.01.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 09.05.03 Bulletin 03/19.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *OBERTHUR CARD SYSTEMS SA Société anonyme* — FR.

(72) **Inventeur(s) :** KNUDSEN ERIK et FEIX BENOIT.

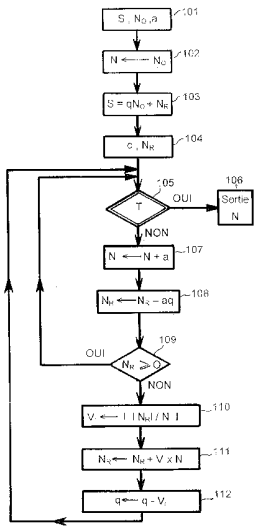
(73) **Titulaire(s) :**

(74) **Mandataire(s) :** CABINET BONNET THIRION.

(54) PROCEDE D'ELABORATION D'UN PARAMETRE DE CRYPTOGRAPHIE.

(57) Détermination d'un nombre premier pour son utilisation dans un processus de cryptographie.

On applique cycliquement un test prédéterminé à un nombre entier (N) qui utilise chaque fois le reste (N_R) et la partie entière du quotient (q) de la division d'un nombre choisi (S) par un nombre entier actuel, à chaque test négatif on sélectionne le nombre entier suivant (N) et on calcule un nouveau reste de la division suivante sans effectuer cette division, directement à partir du reste (N_R) et du quotient (q) et on recommence jusqu'à ce que le test soit positif.



FR 2 832 007 - A1

L'invention concerne un procédé d'élaboration d'un paramètre de cryptographie, typiquement une clé de chiffrement, dans une entité électronique, comme par exemple une carte à microcircuit. L'invention vise plus particulièrement un ensemble d'opérations permettant de déterminer un nombre premier intervenant à un stade du déroulement d'un tel processus cryptographique. Le procédé est remarquable par la rapidité d'exécution des opérations aboutissant à la détermination de ce nombre premier. A titre d'exemple, le procédé peut être mis en œuvre pour accélérer l'élaboration des clés de chiffrement dans un processus dit RSA ou pour initialiser la mise en œuvre d'un processus de type DSA ou Diffie-Hellman.

Dans un processus de détermination d'un nombre premier, on choisit un nombre entier, typiquement un nombre impair assez grand et on lui applique un test, c'est-à-dire un certain algorithme permettant de déterminer si ce nombre est premier ou non. Si le test est négatif, on choisit un autre nombre et on lui ré-applique le même test, et ainsi de suite jusqu'à ce que le résultat de ce test devienne positif. Alors, ce nombre est un nombre premier. Typiquement, les nombres N qui font l'objet du test sont des nombres impairs déduits les uns des autres. Par exemple, on peut passer d'un nombre N au suivant en lui ajoutant un incrément a, de faible valeur. En l'occurrence, par exemple, a = 2. On balaie ainsi une série de nombres entiers impairs en partant d'un premier nombre impair choisi assez grand car les processus cryptographiques font appel à des nombres premiers relativement grands.

Le procédé de l'invention s'applique à tout test capable de déterminer si un nombre choisi est ou non un nombre premier, dès lors qu'il met en œuvre une exponentiation modulaire. Plusieurs tests de ce genre sont connus. On décrira plus particulièrement le procédé de l'invention mis en œuvre avec un test de primalité où on utilise soit l'exponentiation modulaire selon la méthode de Montgomery soit l'exponentiation modulaire selon la méthode de Quisquater.

Pour ce faire, on définit les valeurs suivantes.

Le nombre N, entier est représenté en base b ; il est de longueur n.

Si on pose $R = b^n$; alors $R > N$

On définit le nombre S suivant :

$S = R^2$ si le test met en œuvre la méthode de Montgomery

$S = R \times 2^k$ (k entier petit) si le test met en œuvre la méthode de Quisquater.

Soit la division euclidienne de S par N

on a : $S = q \cdot N + N_R$ avec $0 \leq N_R < N$ (*)

q est le quotient entier de cette division et N_R est le reste. Autrement dit

$N_R = S \bmod N$ et $q = \lfloor S/N \rfloor$

Il est précisé que $\lfloor X \rfloor$ désigne l'entier immédiatement inférieur ou égal à X. Autrement dit, q est toujours un nombre entier et N_R est le reste entier correspondant.

Par ailleurs dans la suite du texte $\lceil X \rceil$ désignera l'entier immédiatement supérieur ou égal à X.

Pour la mise en œuvre de chaque test sur un nombre N de la série définie ou générée comme indiqué ci-dessus, il est nécessaire, en principe de réaliser la division $S = q \cdot N + N_R$

soit $N_R = S \bmod N$.

Si on considère un nombre N ayant donné lieu à un test négatif, le nombre suivant est N + a (typiquement N + 2). On devrait alors être amené à calculer $(N + a)_R = S \bmod (N + a)$. La durée de cette opération est relativement importante.

L'idée de base de l'invention consiste à calculer $(N + a)_R$ directement à partir de N_R, sans effectuer réellement l'opération S mod (N + a), tout en actualisant la valeur de q, le processus étant sensiblement moins coûteux en temps.

Plus précisément, l'invention concerne un procédé d'élaboration d'un paramètre de cryptographie dans une entité électronique, comprenant une phase de détermination d'un nombre premier consistant à appliquer un test prédéterminé à un nombre entier (N) choisi, chaque test utilisant chaque fois le reste (N_R) de la division euclidienne d'un nombre choisi (S) par un nombre entier actuel précité, la partie entière du quotient (q) de cette division étant mémorisée,

caractérisé en ce que, après un test négatif, on sélectionne un nombre entier suivant (N) et on calcule un nouveau reste de la division suivante, sans effectuer cette division, directement à partir dudit reste (N_R) et de ladite partie entière dudit quotient (q), en ce qu'on recommence ledit test et les opérations subséquentes précitées jusqu'à ce que ledit test soit positif et en ce qu'on retient alors le nombre entier correspondant en tant que nombre premier recherché.

Comme indiqué précédemment, on peut générer une série de nombres entiers N au fur et à mesure en incrémentant N d'une valeur entière donnée après chaque test négatif. Autrement dit, on augmente ledit nombre entier N considéré, d'un incrément entier a , petit (typiquement a = 2) et on utilise le résultat en tant que nombre entier suivant précité.

L'invention apparaîtra plus clairement dans la suite de la description et à la lumière des dessins annexés dans lesquels :

- la figure 1 est un organigramme de la mise en œuvre du procédé conforme à l'invention, c'est-à-dire la recherche d'un nombre premier appliquant à chaque nombre choisi un test prédéterminé réalisant une exponentiation modulaire ;

- la figure 2 est une partie d'organigramme illustrant une variante plus générale de l'organigramme de la figure 1 ;

- la figure 3 est un organigramme analogue à celui de la figure 1, lorsque le test fait appel à l'exponentiation selon la méthode de Montgomery ; et

- la figure 4 est un organigramme analogue à celui de la figure 1 lorsque le test fait appel à l'exponentiation selon la méthode de Quisquater.

La justification du procédé énoncé ci-dessus est la suivante.

Si on effectuait la division euclidienne de S par (N + a) on aurait :

$S = q_2 \times (N + a) + (N + a)_R$ avec $(N + a)_R < N + a$

A partir de (*) on peut écrire :

$S = q \times N + N_R + qxa - qxa$

$S = q \times (N + a) + N_R - q \times a$

On calcule donc $(N + a)_R$, le nouveau reste et q_2, le nouveau quotient, de la manière suivante :

On a deux cas possibles :

1. Si $N_R - qxa \geq 0$ alors :

1.1 $q_2 \leftarrow q$

1.2 $(N + a)_R = N_R - a \times q_2$

ou

2. Si $N_R - qxa < 0$ alors

2.1 $(N + a)_R \leftarrow N_R - a \times q$

2.2 $Vi \leftarrow \lceil |(N + a)_R| / (N + a) \rceil$

2.3 $q_2 \leftarrow q - Vi$

2.4 $(N + a)_R \leftarrow (N + a)_R \mod (N + a) = (N + a)_R + Vi \times (N + a)$

Vi est une valeur intermédiaire qu'il est avantageux de calculer en 2.2 pour la suite des opérations 2.3 et 2.4

Il est à noter que la division $|(N + a)_R| / (N + a)$ est une opération qui nécessite peu de temps, parce que, puisque a est petit, $|(N + a)_R|$ et $(N + a)$ sont d'un ordre de grandeur voisin. Ce temps est très inférieur au temps qui serait nécessaire pour effectuer une division telle que S / N.

On en déduit par conséquent en référence à la figure 1, un organigramme général d'un mode d'exécution possible des opérations nécessaires à la détermination d'un nombre premier. Cet organigramme représente une sorte de sous-programme appelé dans le cadre d'un procédé d'élaboration d'un paramètre de cryptographie chaque fois qu'il s'agit de déterminer un nombre premier. Bien entendu, l'organigramme peut être transcrit sous forme de logique câblée dans un microcircuit. Les deux versions étant équivalentes et couvertes par le procédé énoncé ci-dessus.

A l'étape 101, on introduit les données N_0, a, S. N_0 est le premier nombre entier de ladite série, a est la valeur de l'incrément et S est un nombre choisi qui dépend de la nature du test qui sera appliqué, de la base b dans laquelle le nombre N_0 est représenté et de la longueur n de ce nombre.

A l'étape 102, on donne à une mémoire N la valeur N_0.

A l'étape 103, on effectue la division $S = q N_0 + N_R$.

A l'étape 104, on mémorise dans des mémoires q et N_R les valeurs de la partie entière du quotient q et du reste N_R déterminé à l'étape précédente.

L'étape 105 est le test T mettant en œuvre une exponentiation modulaire. Si le test est positif, la valeur inscrite dans le registre mémoire N est un nombre premier. Cette valeur est inscrite en sortie et le processus de détermination du nombre premier est terminé.

Si le test est négatif, on passe à l'étape 107 où la valeur inscrite dans le registre mémoire N est augmentée de l'incrément a.

A l'étape 108, on retranche au reste N_R précédant, le produit dudit incrément a par ladite partie entière du quotient q et on réinscrit le résultat dans la mémoire N_R.

L'étape 109 est un test où l'on détermine si N_R est positif ou nul. Si la réponse est oui, on retient le nouveau reste $(N + a)_R$ déterminé à l'étape 108 et on retourne à l'étape 105 pour réappliquer le test T sans modifier ladite partie entière du quotient q. Si le test 109 est négatif, on passe à l'étape 110 où on calcule une valeur intermédiaire Vi (mémorisée dans une mémoire Vi) en prenant l'entier immédiatement supérieur (ou égal si $N_R = 0$) à la partie entière du quotient de la division de la valeur absolue du nombre obtenu à l'étape 108 (mémorisé dans la mémoire N_R) par ledit nombre entier suivant N calculé et mémorisé à l'étape 107. On passe alors à l'étape 111 où on ajoute au nombre obtenu à l'étape 108 le produit dudit nombre entier suivant, obtenu à l'étape 107 par ladite valeur intermédiaire Vi et on retient cette somme en tant que nouveau reste $(N + a)_R$ que l'on inscrit dans la mémoire N_R.

On passe ensuite à l'étape 112 où on retranche ladite valeur intermédiaire Vi à ladite partie entière du quotient q et on retient cette différence en tant que nouvelle partie entière du quotient q, en l'inscrivant dans la mémoire q. Après l'étape 112, on revient au test T de l'étape 105 et on effectue les mêmes opérations jusqu'à ce que le test T soit positif. Chaque test T est ainsi opéré à partir d'un nombre entier N actuel déterminé à l'étape 107.

La figure 2 illustre une variante dans laquelle le nombre N n'est pas déduit du précédent. Les opérations successives 107a, 107b et 107c se substituent à l'opération 107 de la figure 1. Il n'est pas nécessaire d'initialiser $\underline{a}$ à l'étape 101.

- A l'étape 107a on inscrit temporairement dans une mémoire N_p la valeur actuelle de N.

- A l'étape 107b on choisit une autre valeur de N sans que la nouvelle valeur de N soit nécessairement déduite de la précédente. Par exemple, N peut être choisi de façon aléatoire dans un intervalle de nombres tel que a reste petit devant N.

- A l'étape 107c on calcule la valeur de a comme étant la différence entre N et N_p. On peut compléter cette étape par un test sur a permettant de vérifier que a est effectivement petit devant N. Ainsi, comme mentionné précédemment, la division de l'étape 110 prendra peu de temps. Dans la pratique, on a vu qu'on pouvait choisir a=2 (en partant d'un nombre N impair) ce qui assure de trouver le plus petit nombre premier supérieur à N_0 . Dans ce cas chaque nombre n est déduit du précédent à l'étape 107 de la figure 1. Cependant, le procédé donne des résultats satisfaisants tant que la division de l'étape 110 peut être effectuée en un temps court. On estime ainsi que ce sera le cas tant que $\underline{a}$ reste inférieur à N/10, cette limite n'étant cependant nullement impérative.

La figure 3 illustre la détermination d'un nombre premier N conforme au principe décrit ci-dessus en référence à la figure 1 mais dans les conditions particulières qui résultent de l'utilisation de la méthode d'exponentiation de Montgomery dans le test, noté T_M pour cette raison à l'étape 105. Dans l'organigramme de la figure 3, les étapes analogues à celles de la figure 1 portent les mêmes références et ne seront pas décrites à nouveau. On constate que la seule différence réside dans le fait que l'étape 109 de la figure 1 peut être supprimée lorsqu'on utilise l'exponentiation de Montgomery.

En effet, on rappelle que le test de primalité de Fermat permettant de déterminer si N est premier, consiste à choisir un nombre x compris entre 2 et N - 1 et à calculer l'expression x^{N-1} mod N.

Si cette valeur est différente de 1 le test est négatif et si cette valeur est égale à 1 le test est positif, c'est-à-dire que le nombre N est premier.

Dans le cas où on utilise l'exponentiation de Montgomery, on a choisi

$$S = R^2.$$

De plus, comme

$$R > N$$

$$S = R^2 > N^2$$

$$S = q \cdot N + N_R > N^2 \text{ avec } 1 < N_R < N$$

$$q \cdot N > N^2 - N_R$$

$$q > N - (N_R / N)$$

or $N_R < N$

donc $0 \leq N_R / N < 1$

d'où $q \geq N$ parce que q est entier

Comme en outre $N_R < N$, on a :

$$N_R - q \times a < 0$$

Ce qui démontre bien que le test 109 est inutile dans le cas de l'utilisation de l'exponentiation de Montgomery.

La figure 4 illustre la détermination d'un nombre premier N conforme au principe général décrit en référence à la figure 1 mais dans les conditions particulières qui résultent de l'utilisation de la méthode d'exponentiation de Quisquater dans le test, noté T_Q pour cette raison à l'étape 105. Dans l'organigramme de la figure 4, les étapes analogues à celles de la figure 1 portent les mêmes références et ne seront pas décrites à nouveau. On constate que les seuls traitements différents concernent les opérations effectuées après l'étape 109 dans le cas où la réponse à ce test est négative.

On rappelle que dans le cas de la méthode d'exponentiation de Quisquater

$$S = R \times 2^k \text{ avec k petit.}$$

Pour simplifier les opérations, on recherche une condition sur la valeur de a.

Plus précisément, on a besoin de la propriété suivante :

$$N_R - aq \geq -(N + a)$$

soit $a \leq (N + N_R) / (q - 1)$

or $S = q \times N + N_R$

d'où $q = (S - N_R) / N$

on obtient alors :

$$a \leq N \times (N + N_R) / (S - N_R - N)$$

comme a est un entier petit (typiquement égal à 2) cette condition peut être considérée comme remplie dans tous les cas.

Par ailleurs, si on effectuait la division euclidienne de S par (N + a), on aurait :

$$S = q_2 \times (N + a) + (N + a)_R$$

avec $0 \leq (N + a)_R < N + a$

comme précédemment on peut écrire :

$$S = q \times N + N_R + q \times a - q \times a$$

$$S = q \times (N + a) + N_R - q \times a$$

or, compte tenu de la condition sur a :

$$(N + a) > N_R - q \times a \geq -(N + a)$$

On a donc deux cas possibles :

1. Si $N_R - q \times a \geq 0$ alors :

   1.1 $q = q_2$

   1.2 $(N + a)_R = N_R - a \times q_2$

2. Si $N_R - q \times a < 0$ alors :

   2.1 $q - 1 = q_2$

   2.2 $(N + a)_R = N_R - a \times q + N + a$

Ces deux cas se transcrivent par les opérations qui suivent un test T_Q négatif.

Les opérations 107 et 108 ne changent pas par rapport à l'organigramme de la figure 1 ; elles correspondent à l'incrémentation de N et à une première modification du contenu de la mémoire N_R, suffisante lorsque $N_R - q \times a$ est positif ou nul. Dans ce cas, cette valeur est retenue en tant que nouveau reste

$(N + a)_R$ et inscrite dans la mémoire N_R. La valeur inscrite dans la mémoire q n'est pas modifiée.

Dans la pratique, la condition

$$N_1 - q \times a \geq 0$$

est presque toujours réalisée ce qui confirme que les opérations 105 à 108 peuvent être exécutées très rapidement.

Il peut se produire cependant que le test 109 soit négatif. Par conséquent, si le nombre obtenu et mémorisé à l'étape 108 est négatif, on lui ajoute ledit nombre entier suivant (N + a) obtenu à l'étape 107 et mémorisé dans la mémoire N et on retient la somme $N_R + N$ en tant que nouveau reste $(N + a)_R$, soit l'étape 113 :

$$N_R \leftarrow N_R + N$$

Par ailleurs, à l'étape 114, on retranche 1 à la partie entière du quotient q et on retient cette différence en tant que nouvelle partie entière du quotient en l'inscrivant dans la mémoire q. On retourne ensuite à l'étape 105 pour l'application d'un nouveau test de primalité T_Q utilisant l'exponentiation de Quisquater.

Les principales applications du procédé de détermination décrit ci-dessus sont les suivantes.

De façon générale un tel procédé de génération de nombre premier peut être utilisé pour la génération de clés dans tout système cryptographique à clé publique.

Dans un système à clé publique, chaque utilisateur publie une clé qui permet à n'importe quel utilisateur du système de lui envoyer un message crypté. Parallèlement, il garde secrète une clé privée associée qui lui permet de décrypter tout message crypté avec cette clé publique par un autre utilisateur. La clé privée peut également être utilisée pour signer un message ou réaliser une authentification.

Pour chaque utilisateur, la clé publique et la clé privée associée sont engendrées à partir de nombres premiers. Ces clés peuvent être créées une fois pour toutes pour chaque nouvel utilisateur, ou renouvelées plus ou moins

fréquemment. Pour assurer la confidentialité des clés, les nombres premiers engendrés doivent être aléatoires, ce que permet le procédé.

Dans les applications avec carte à microcircuit, les clés publiques sont créées, en général par le fabricant de cartes, durant la phase de personnalisation, en utilisant des systèmes électroniques dédiés à cette tâche. Les cartes sont alors commandées par lots par l'organisme qui délivre la carte (banque, administration...). La commande réalisée par cet organisme est alors accompagnée des informations nécessaires à la personnalisation des cartes (nom des porteurs, numéro de compte...). Pour les petits volumes de cartes notamment, c'est parfois la carte elle-même qui est capable de créer les clés publique et privée grâce à un algorithme de génération de nombre premier aléatoire intégré dans le microcircuit de la carte. L'organisme qui délivre la carte peut alors lui-même réaliser la personnalisation de la carte. Sur une carte à microcircuit, une telle génération de clé peut demander trente secondes de calcul. On comprend alors la nécessité de réduire ces temps de calcul. On peut avoir besoin de délivrer un nombre important de cartes, par exemple un ou plusieurs milliers de cartes.

La personnalisation de la carte peut être faite :

- par un serveur qui possède alors des moyens de détermination d'un nombre premier selon le procédé décrit ci-dessus, la clé étant ensuite adressée à la carte et mémorisée dans celle-ci.
- par la carte elle-même si le microcircuit qu'elle renferme possède des moyens de détermination d'un nombre premier selon le procédé décrit, la carte opérant cette personnalisation sur ordre.

Le fait de générer une clé secrète complètement dans la carte est avantageux. On améliore la sécurité car la clé ainsi déterminée ne "sort" jamais de la carte.

Par conséquent, l'invention couvre aussi toute entité informatique comme par exemple un serveur ou une carte à microcircuit dès lors qu'elle est équipée de moyens de mise en œuvre (logiciels, progiciels ou autres) du procédé décrit ci-dessus.

REVENDICATIONS

1. Procédé d'élaboration d'un paramètre de cryptographie dans une entité électronique, comprenant une phase de détermination d'un nombre premier consistant à appliquer un test prédéterminé à un nombre entier (N) choisi, chaque test utilisant chaque fois le reste (N_R) de la division euclidienne d'un nombre choisi (S) par un nombre entier actuel précité, la partie entière du quotient (q) de cette division étant mémorisée, caractérisé en ce que, après un test négatif, on sélectionne un nombre entier suivant (N) et on calcule un nouveau reste de la division suivante, sans effectuer cette division, directement à partir dudit reste (N_R) et de ladite partie entière dudit quotient (q), en ce qu'on recommence ledit test et les opérations subséquentes précitées jusqu'à ce que ledit test soit positif et en ce qu'on retient alors le nombre entier correspondant en tant que nombre premier recherché.

2. Procédé selon la revendication 1, caractérisé en ce qu'on choisit ledit nombre entier suivant de telle sorte que la différence entre ledit nombre entier suivant et ledit nombre entier actuel soit égale à un nombre entier (a) petit.

3. Procédé selon la revendication 1, caractérisé en ce que les nombres entiers auxquels on applique successivement ledit test sont déduits les uns des autres.

4. Procédé selon la revendication 3, caractérisé en ce qu'on augmente ledit nombre entier (N) considéré, d'un incrément entier (a), petit, et en ce qu'on utilise le résultat en tant que nombre entier suivant précité.

5. Procédé selon la revendication 4, caractérisé en ce qu'on retranche au reste considéré le produit dudit incrément (a) par ladite partie entière dudit quotient (q), en ce que, si le nombre obtenu est positif ou nul, on le retient en tant que nouveau reste ($(N+a)_R$) pour le test suivant sans modifier ladite partie entière (q) dudit quotient, ou en ce que si ledit nombre obtenu est négatif, on calcule une valeur intermédiaire (Vi) égale à l'entier immédiatement supérieur ou égal à la partie entière du quotient de la division de la valeur absolue dudit nombre obtenu par ledit nombre entier suivant (N), en ce qu'on ajoute audit nombre obtenu le produit dudit nombre entier suivant par ladite valeur

intermédiaire, en ce qu'on retient cette somme en tant que nouveau reste $((N+a)_R)$, en ce qu'on retranche ladite valeur intermédiaire (Vi) à ladite partie entière (q) et en ce qu'on retient cette différence en tant que nouvelle partie entière (q).

6. Procédé selon la revendication 4, dans lequel chaque nombre entier (N) de ladite série est de longueur n représenté en base b, caractérisé en ce que ledit test comporte une exponentiation modulaire effectuée par la méthode de Montgomery, ledit nombre choisi (S) étant égal au carré de b à la puissance n.

7. Procédé selon la revendication 6, caractérisé en ce qu'on retranche au reste considéré le produit dudit incrément (a) par ladite partie entière dudit quotient (q), en ce qu'on calcule une valeur intermédiaire (Vi) égale à l'entier immédiatement supérieur ou égal à la partie entière de la division de la valeur absolue dudit nombre obtenu par ledit nombre entier suivant (N), en ce qu'on ajoute audit nombre obtenu le produit dudit nombre entier suivant par ladite valeur intermédiaire, en ce qu'on retient cette somme en tant que nouveau reste $((N+a)_R)$, en ce qu'on retranche ladite valeur intermédiaire (Vi) à ladite partie entière (q), et en ce qu'on retient cette différence en tant que nouvelle partie entière (q).

8. Procédé selon la revendication 4, dans lequel chaque nombre entier (N) de ladite série est de longueur n représenté en base b, caractérisé en ce que ledit test comporte une exponentiation modulaire par la méthode de Quisquater, ledit nombre choisi (S) étant égal au produit de b^n par 2^k , k étant un entier petit.

9. Procédé selon la revendication 8, caractérisé en ce qu'on retranche au reste considéré le produit dudit incrément (a) par ladite partie entière dudit quotient (q), en ce que, si le nombre obtenu est positif ou nul, on le retient en tant que nouveau reste $((N+a)_R)$, pour le test suivant sans modifier ladite partie entière (q) dudit quotient ou en ce que, si ledit nombre obtenu est négatif, on lui ajoute ledit nombre entier suivant (N+a) et on retient cette somme en tant que nouveau reste $((N+a)_R)$, en ce qu'on retranche 1 à ladite partie entière (q), et en ce qu'on retient cette différence en tant que nouvelle partie entière (q).

10. Entité informatique caractérisée en ce qu'elle est équipée de moyens de mise en œuvre du procédé selon l'une des revendications précédentes.

11. Entité informatique selon la revendication 10, caractérisée en ce qu'elle constitue une carte à microcircuit.

Fig. 1

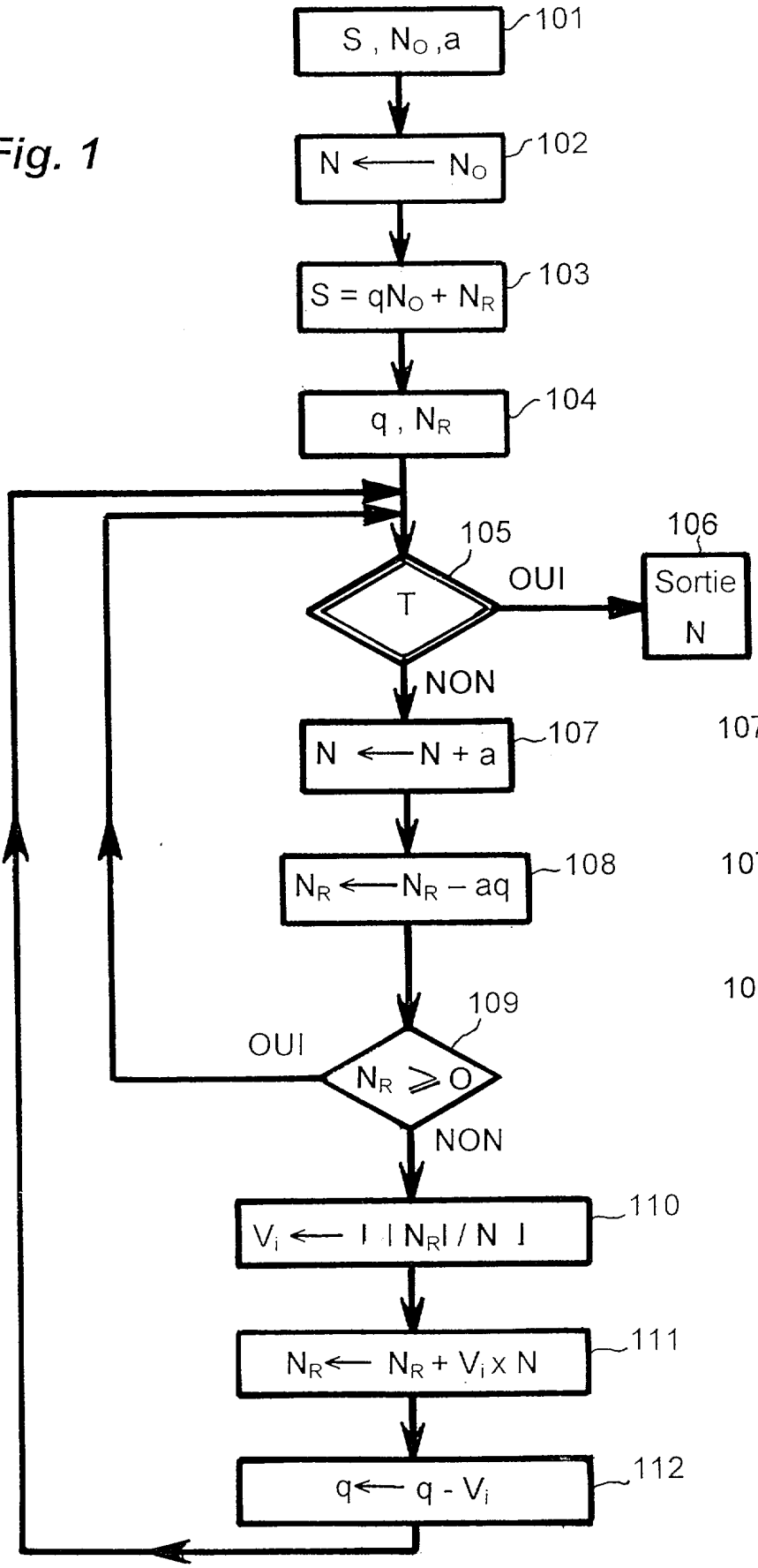

S , N$_O$,a
101
N ⟵ N$_O$
102
S = qN$_O$ + N$_R$
103
q , N$_R$
104
105
T
OUI
106
Sortie N
NON
N ⟵ N + a
107
N$_R$ ⟵ N$_R$ – aq
108
109
OUI
N$_R$ ⩾ O
NON
V$_i$ ⟵ I I N$_R$I / N I
110
N$_R$⟵ N$_R$ + V$_i$ x N
111
q⟵ q - V$_i$
112

Fig. 2

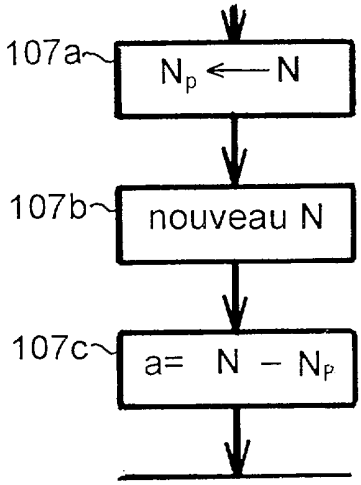

107a
N$_p$ ⟵ N
107b
nouveau N
107c
a= N – N$_P$

Fig. 3

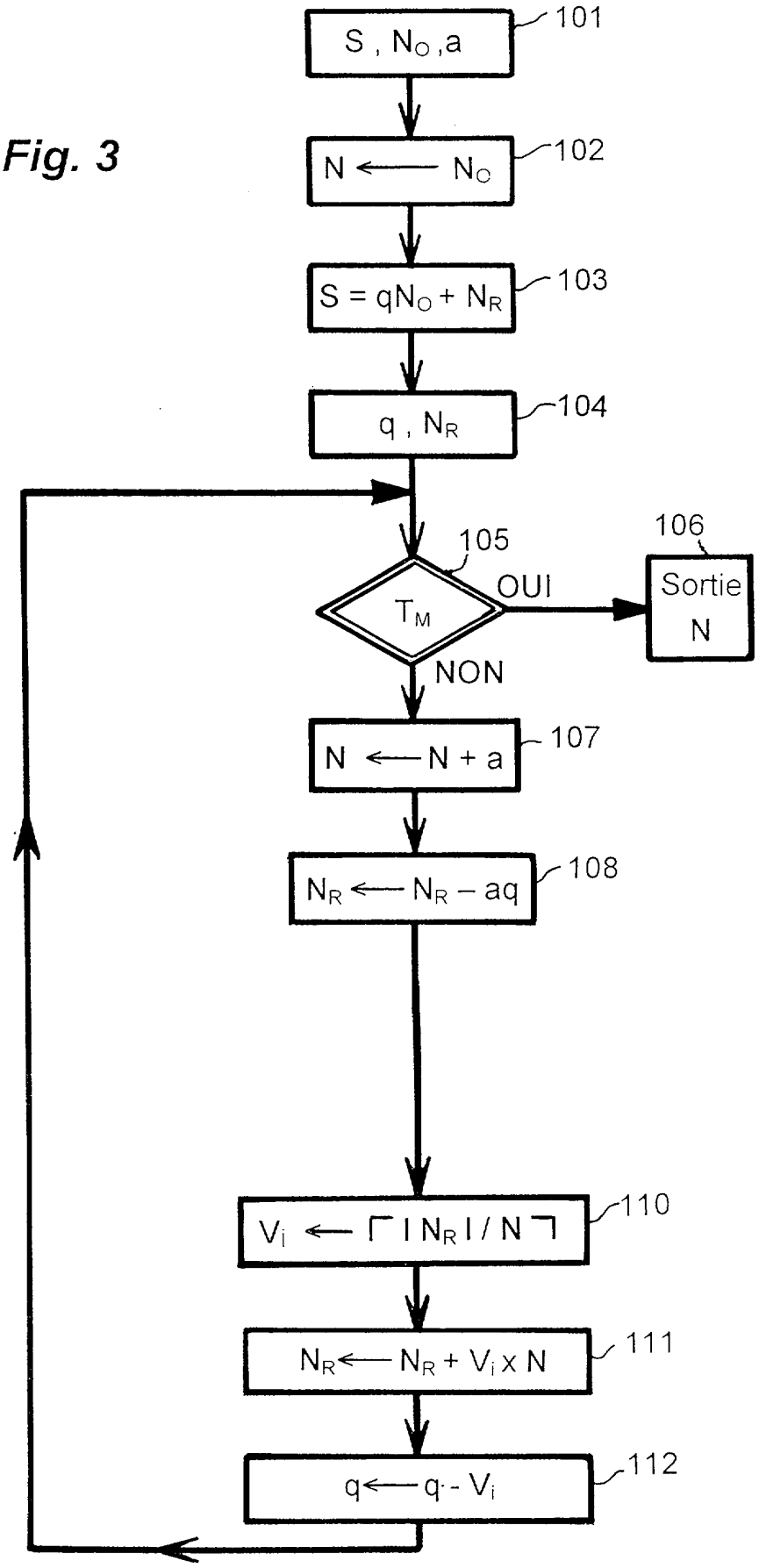
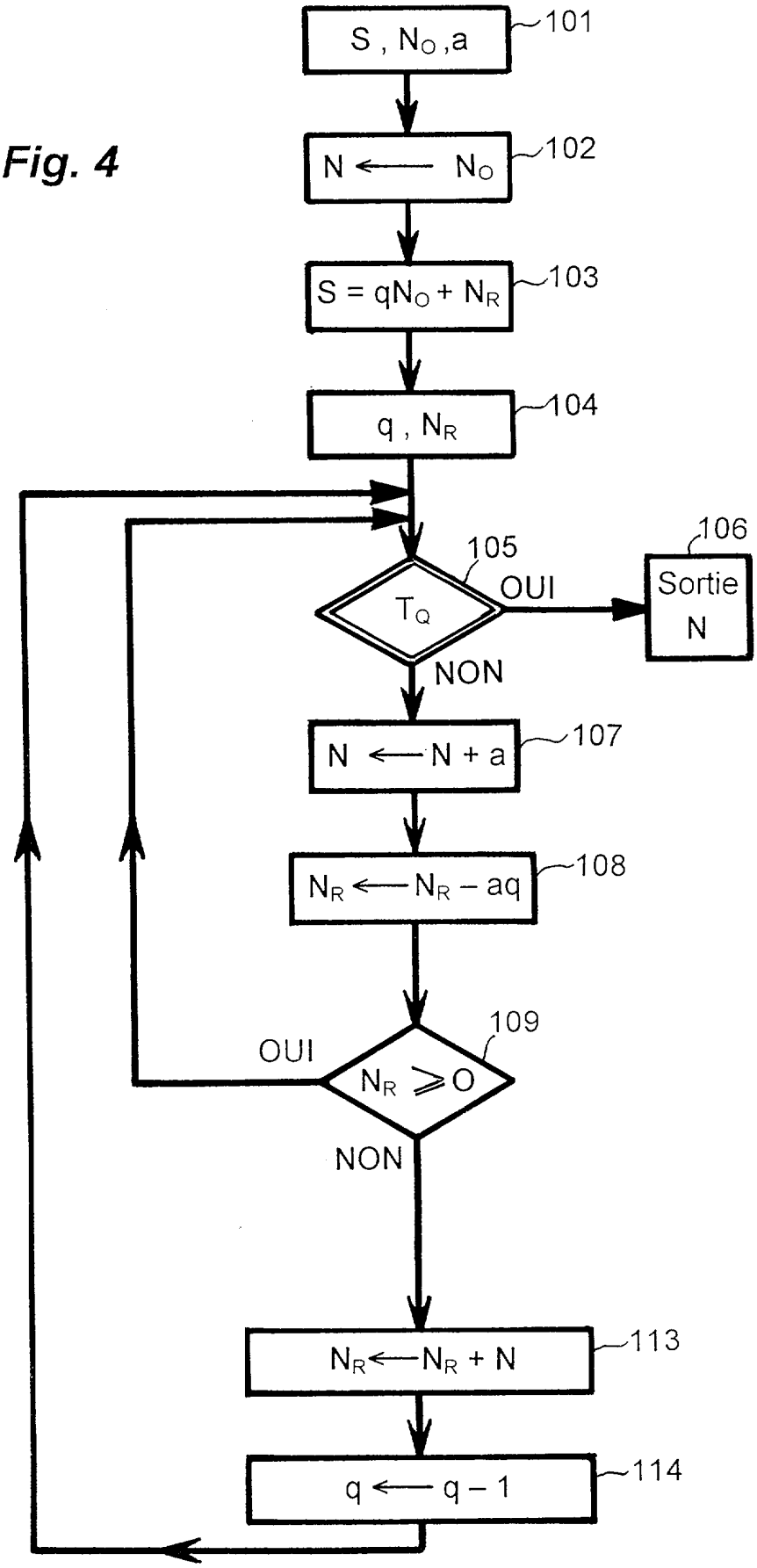

S , N$_O$,a
101
N ⟵ N$_O$
102
S = qN$_O$ + N$_R$
103
q , N$_R$
104
105
T$_M$
OUI
106
Sortie N
NON
N ⟵ N + a
107
N$_R$ ⟵ N$_R$ – aq
108
V$_i$ ⟵ ⌈ I N$_R$ I / N ⌉
110
N$_R$⟵ N$_R$ + V$_i$ x N
111
q⟵ q - V$_i$
112

Fig. 4


S , N_O ,a
101
N ⟵ N_O
102
S = $qN_O + N_R$
103
q , N_R
104
105
T_Q
OUI
106
Sortie N
NON
N ⟵ N + a
107
N_R ⟵ $N_R - aq$
108
109
OUI
$N_R \geqslant O$
NON
N_R⟵ N_R + N
113
q ⟵ q – 1
114

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

2832007
N° d'enregistrement national

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

FA 612334
FR 0114272

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	J-F DHEM: "Design of an efficient public-key cryptographic library for RISC-based smart cards" , UNIVERSITÉ CATHOLIQUE DE LOUVAIN , LOUVAIN-LA-NEUVE XP002207928 * page 11, alinéa 1 - page 29, alinéa 6 * * page 105, alinéa 3 - page 120, alinéa 5 * ---	1-11	H04L9/18
A	ARAZI B: "ON PRIMALITY TESTING USING PURELY DIVISIONLESS OPERATIONS" COMPUTER JOURNAL, OXFORD UNIVERSITY PRESS, SURREY, GB, vol. 37, no. 3, 1994, pages 219-222, XP000448174 ISSN: 0010-4620 * le document en entier * ---	1,6,7	
A	MONTGOMERY P L: "MODULAR MULTIPLICATION WITHOUT TRIAL DIVISION" MATHEMATICS OF COMPUTATION, AMERICAN MATHEMATICAL SOCIETY, US, vol. 44, no. 170, 1 avril 1985 (1985-04-01), pages 519-521, XP000747434 * le document en entier * ---	6	DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7) H04L G06F
A	EP 0 350 278 A (BRITISH AEROSPACE ;HATFIELD POLYTECHNIC (GB)) 10 janvier 1990 (1990-01-10) * abrégé * * page 3, ligne 52 - ligne 58 * -----	1	

Date d'achèvement de la recherche	Examinateur
9 août 2002	Dujardin, C

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0114272 FA 612334

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 09-08-2002
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP 0350278 A	10-01-1990	EP 0350278 A2	10-01-1990
		US 5121429 A	09-06-1992

EPO FORM P0465