(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 873 523**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **04 08139**

(51) Int Cl[8] : **H 04 L 9/30** (2006.01), H 04 L 9/06

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 22.07.04.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 27.01.06 Bulletin 06/04.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Ce dernier n'a pas été établi à la date de publication de la demande.*

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

(71) **Demandeur(s) :** *SAGEM SA Société anonyme* — FR.

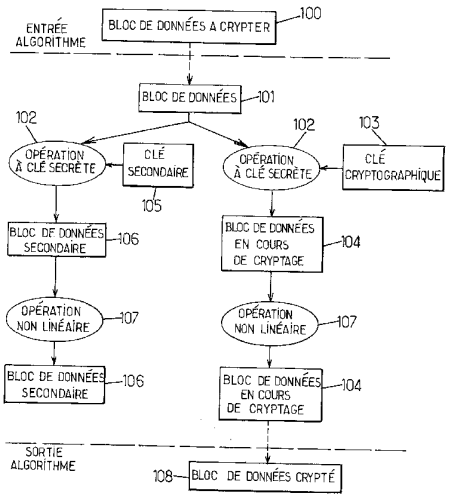
(72) **Inventeur(s) :** PELLETIER HERVE.

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

(74) **Mandataire(s) :** CABINET PLASSERAUD.

(54) PROCEDE ET DISPOSITIF D'EXECUTION D'UN CALCUL CRYPTOGRAPHIQUE.

(57) On exécute un calcul cryptographique dans un composant électronique, selon un algorithme cryptographique déterminé incluant au moins une opération à clé secrète (102) à réaliser avec une clé cryptographique secrète (103) comprenant m blocs de clé cryptographique secrète de n bits sur un bloc de données (101), où m et n sont des entiers positifs, et une opération non linéaire (107). On détermine au moins une clé secrète secondaire (105) sur n bits différente du bloc de clé cryptographique secrète. Puis, pour un bloc de clé cryptographique secrète donné, on réalise l'opération à clé secrète (102) avec le bloc de clé cryptographique secrète (103) et on réalise l'opération à clé secrète (102) avec la clé secrète secondaire sur un bloc de données (101) et on obtient respectivement un bloc de données en cours de cryptage (104) et un bloc de données secondaire (106). Ensuite, on réalise l'opération non linéaire (107) sur le bloc de données en cours de cryptage (104) et sur le bloc de données secondaire (106). On fournit un bloc de données crypté à partir du bloc de données en cours de cryptage.



FR 2 873 523 - A1

PROCEDE ET DISPOSITIF D'EXECUTION D'UN CALCUL CRYPTOGRAPHIQUE

La présente invention est relative au domaine de la cryptographie et plus particulièrement à la protection de la confidentialité des clés utilisées par des algorithmes cryptographiques.

Les algorithmes cryptographiques ont pour objectif de crypter des données. De tels algorithmes comprennent généralement un enchaînement de plusieurs opérations, ou calculs, que l'on applique successivement sur une donnée à crypter afin d'obtenir une donnée cryptée. Ces algorithmes utilisent des clés secrètes.

De tels algorithmes cryptographiques peuvent subir des 'attaques' qui visent à violer la confidentialité de clés utilisées. De nombreux types d'attaques sont aujourd'hui connus.

Ainsi, certaines attaques sont fondées sur des fuites d'information détectées lors de l'exécution de l'algorithme de chiffrement. Elles sont généralement basées sur une corrélation entre les fuites d'informations détectées lors du traitement par l'algorithme de chiffrement de la donnée et de la clé ou des clés secrètes utilisées. On connaît ainsi des attaques DPA pour 'Differential Power Analysis' en anglais. Ces dernières requièrent en général une connaissance des données de sortie cryptées. On connaît également des attaques SPA pour 'Simple Power Analysis' basées sur une analyse d'un simple graphe de consommation de puissance comme cela est décrit dans le document 'Smartly analyzing the simplicity and the power of simple power analysis on Smartcards', Rita Mayer-Sommer electrical engineering division ETH Zürich, 2000.

Un algorithme cryptographique comprend de manière générale plusieurs opérations linéaires et/ou non linéaires. Pour une donnée initiale à crypter, on obtient une donnée intermédiaire en cours de cryptage après chacune des opérations de l'algorithme.

Ainsi, un algorithme de type DES pour 'Data Encryption Standard' ou encore l'algorithme AES pour 'Advanced Encryption Standard' comprend des

opérations non linéaires. Les attaques DPA et SPA s'avèrent être particulièrement pertinentes contre l'algorithme AES lors de l'exécution des opérations non linéaires.

Plusieurs procédés de protection d'algorithmes cryptographiques de ce type ont déjà été proposés, notamment par masquage des données en cours de cryptage manipulées dans l'algorithme AES. Les opérations non linéaires sont généralement implémentées sous forme de tables de substitution. Ainsi, une opération non linéaire correspondant à une table de substitution tab[i], appliquée à une donnée x peut s'écrire sous la forme suivante :

y= tab[x].

Il est parfois complexe de masquer une donnée au cours d'une opération non linéaire avec un masque de valeur aléatoire.

La présente invention vise à proposer une méthode facile à implémenter pour protéger efficacement des exécutions de calculs des algorithmes cryptographiques basés sur au moins une clé secrète contre des attaques DPA ou encore SPA.

Un premier aspect de l'invention propose un procédé d'exécution d'un calcul cryptographique dans un composant électronique, selon un algorithme cryptographique déterminé incluant au moins une opération à clé secrète à réaliser avec une clé cryptographique secrète comprenant m de blocs de clé cryptographique secrète de n bits sur un bloc de données, où m et n sont des nombres entiers positifs. Le procédé comprend, pour un bloc de clé cryptographique secrète donné, les étapes suivantes consistant à :

- déterminer au moins une clé secrète secondaire sur n bits différente dudit bloc de clé cryptographique secrète ;
- réaliser ladite opération à clé secrète avec ledit bloc de clé cryptographique secrète et réaliser ladite opération à clé secrète avec ladite clé secrète secondaire sur un bloc de données et obtenir respectivement un bloc de données en cours de cryptage et un bloc de données secondaire;
- réaliser ladite opération non linéaire sur ledit bloc de données en cours de cryptage et sur ledit bloc de données secondaire ;
- fournir un bloc de données crypté à partir du bloc de données en cours de cryptage.

On note que les clés secrètes secondaires sont des clés factices.

Grâce à ces dispositions, on génère, outre les fuites d'information liées aux calculs cryptographiques exécutés sur un bloc de données en cours de cryptage, des fuites d'information liées aux calculs cryptographiques exécutés sur un bloc de données secondaire. Une analyse de telles fuites d'information est de ce fait plus complexe et prend donc plus de temps que l'analyse de fuites d'information lors de l'exécution de calculs cryptographiques sur le bloc de données en cours de cryptage uniquement. On protège ainsi la confidentialité des clés cryptographiques secrètes. La complexité d'une attaque d'un tel algorithme augmente avec le nombre de fois où l'on réalise l'opération 102 à clé secrète avec une clé secrète secondaire factice différente.

Un tel procédé vise à faire apparaître des biais de corrélation pour rendre les attaques SPA ou DPA plus longues voire impossibles.

Ainsi, dans un mode de réalisation de la présente invention, afin de garantir une meilleure confidentialité de l'algorithme, pour un bloc de clé cryptographique secrète, on détermine toutes les valeurs possibles de clés sur n bits, c'est-à-dire 2^n valeurs, ou encore 2^n-1 clés secrètes secondaires factices sur n bits différentes du bloc de clé cryptographique secrète. Puis, on réalise l'opération à clé secrète avec la clé cryptographique secrète et également toutes ces clés secrètes secondaires déterminées. On obtient alors un bloc de données en cours de cryptage et 2^n-1 blocs de données secondaires sur lesquels on réalise l'opération non linéaire. Dans ce cas, les clés cryptographiques secrètes ne sont pas détectables.

Dans un mode de réalisation de l'invention, le procédé comprend, pour un bloc de clé cryptographique secrète, les étapes consistant à:

- déterminer et arranger aléatoirement un nombre déterminé p de clés secrètes dans une table initiale comprenant ledit bloc de clé cryptographique secrète;
- stocker en mémoire l'adresse correspondant audit bloc de clé cryptographique secrète dans la table initiale;
- appliquer l'opération à clé secrète au bloc de données avec les clés de la table initiale et obtenir une première table transformée de p premiers éléments, chaque premier élément correspondant au résultat de l'opération

à clé secrète appliquée au bloc de données avec la clé située dans la table initiale à la même adresse que ledit premier élément;

- appliquer ladite opération non linéaire aux éléments de ladite première table transformée et obtenir une seconde table transformée de p seconds éléments, chaque second élément correspondant au résultat de l'opération non linéaire appliquée au premier élément situé à la même adresse dans la première table transformé que ledit second élément;
- récupérer, dans la seconde table transformée, l'élément correspondant au bloc de données en cours de cryptage situé à l'adresse dudit bloc de clé cryptographique secrète.

On peut avantageusement choisir le nombre p de clés secrètes dans la table initiale égal à 2^n. Dans ce cas, la table initiale comprend toutes les valeurs de clé possibles.

Grâce à ces dispositions, on manipule de manière équiprobable toutes les clés possibles quand on exécute l'opération à clé secrète et l'opération non linéaire. Un tel procédé garantit une très grande protection de la confidentialité de l'algorithme quant aux attaques DPA et SPA.

On peut récupérer l'élément correspondant au bloc de données en cours de cryptage dans la seconde table transformée via une fonction SPA résistante prenant en paramètre l'adresse du bloc de clé cryptographique secrète donné. On entend par "fonction résistante contre les attaques de type SPA' une fonction pour laquelle il n'est pas possible de déterminer une clé secrète en une seule trace de fuite. En considérant que le signal de fuite W, correspondant à un courant, ou encore à un champ électromagnétique, lors d'une manipulation d'un octet α pour un calcul de l'algorithme, est de la forme suivante :

$$W(\alpha)=H(\alpha)+b ;$$

où $H(\alpha)$ est le modèle de fuite et b le bruit extrinsèque et intrinsèque.

On considère qu'une fonction exécutée sur l'octet α est une fonction résistante contre les attaques SPA lorsqu'on a l'équation suivante :

$$|W_\alpha - W_{\alpha'}| \leq b ,$$

où α' est un autre octet.

Lorsque l'algorithme cryptographique comprend un nombre déterminé de tours, chacun incluant au moins une opération à clé cryptographique secrète précédant une opération non linéaire réalisée via une table de substitution, on peut réaliser les étapes du procédé énoncées ci-dessus pour au moins le premier tour et au moins le dernier tour de l'algorithme cryptographique.

En effet, les premiers et derniers tours de l'algorithme AES sont les plus fragiles face aux attaques de type SPA et DPA. Ainsi, en appliquant le procédé selon un mode de réalisation de la présente invention au premier tour et au dernier tour, on protège la confidentialité de l'algorithme tout en limitant le nombre de calculs à ajouter pour la protection de cet algorithme.

L'étape d'arrangement aléatoire des clés dans la table peut être fait à chaque début de l'algorithme cryptographique.

Par ailleurs, on peut réaliser simultanément l'opération à clé secrète avec un bloc de clé cryptographique secrète et avec la clé secrète secondaire et/ou on peut réaliser simultanément l'opération non linéaire sur le bloc de données et les blocs de données secondaires afin de fournir de bonne performance quant à l'exécution des calculs de l'algorithme.

Dans un mode de réalisation de la présente invention, l'algorithme cryptographique est l'AES.

Dans un mode de réalisation de la présente invention, l'une au moins des opérations de l'algorithme cryptographique est réalisée sur le bloc de données en cours de cryptage masqué avec une valeur aléatoire. De préférence, les opérations de l'algorithme autres que celles réalisées avec au moins une clé secrète secondaire et celles réalisées sur des blocs de données secondaires, sont réalisées sur un bloc de données en cours de cryptage qui est masqué

Un autre aspect de l'invention propose un composant électronique adapté pour exécuter un calcul cryptographique selon un algorithme cryptographique déterminé incluant au moins une opération à clé secrète à réaliser avec une clé cryptographique secrète comprenant m blocs de clé secrète de n bits sur un bloc de données et une opération non linéaire, comprenant des moyens agencés pour mettre en œuvre un procédé comme

énoncé ci-dessus.

D'autres aspects, buts et avantages de l'invention apparaîtront à la lecture de la description d'un de ses modes de réalisation.

L'invention sera également mieux comprise à l'aide des dessins, sur lesquels :

- la figure 1 illustre un procédé de calcul cryptographique selon un mode de réalisation de la présente invention ;
- la figure 2 illustre les principales étapes d'un algorithme de type AES ;
- la figure 3 illustre une opération à clé secrète selon un mode de réalisation de l'invention ;
- la figure 4 illustre l'exécution d'une opération non linéaire selon un mode de réalisation de la présente invention ;
- la figure 5 illustre une gestion du passage au premier tour d'un algorithme de type AES comprenant des calculs cryptographiques exécutés selon un mode de réalisation de la présente invention ;
- la figure 6 illustre une gestion du passage entre deux tours consécutifs d'un algorithme de type AES comprenant des calculs cryptographiques exécutés selon un mode de réalisation de la présente invention.

Généralement, un algorithme cryptographique comprend plusieurs opérations qui sont appliquées successivement à un bloc de données, chacune étant appliquée au bloc de données transformé par l'opération précédente. En sortie de l'algorithme, un bloc de données en cours de cryptage est un bloc de données crypté.

La figure 1 illustre un procédé d'exécution d'un calcul cryptographique selon un algorithme cryptographique, selon un mode de réalisation de la présente invention. Un tel algorithme inclut au moins une opération à clé secrète 102 à réaliser avec une clé cryptographique secrète 103 sur un bloc de données 101 pour obtenir un bloc de données en cours de cryptage 104. L'algorithme inclut également une opération non linéaire 107 à réaliser sur le bloc de données en cours de cryptage 104 pour obtenir un autre bloc de données en cours de cryptage 104. Le bloc de données 101 peut être le résultat d'une opération précédente dans le cas où une ou plusieurs opérations précèdent l'opération à clé secrète 102. Dans le cas où l'opération à clé secrète

102 est la première opération de l'algorithme, il peut correspondre au bloc de données à crypter 100, reçu en entrée de l'algorithme.

A titre d'exemple, dans le cas où l'opération à clé secrète est réalisée avec une clé cryptographique secrète de 128 bits comprenant 16 blocs de clé cryptographique secrète d'un octet chacun, l'opération à clé secrète 102 est réalisée 16 fois sur un bloc de données d'un octet, une fois avec chacun des blocs de clé cryptographique secrète. Après avoir déterminé une valeur de clé secrète secondaire 105 différente de la valeur du bloc de clé cryptographique secrète 103 correspondant, on réalise l'opération 102 avec la clé secrète secondaire 105 déterminée sur le bloc de données 101 pour obtenir un bloc de données secondaire 106. Puis, on applique sur ce bloc de données secondaire, l'opération non linéaire 107 pour obtenir un autre bloc de données secondaire 106.

En sortie de l'algorithme cryptographique on obtient un bloc de données crypté 108.

L'invention couvre toutes les implémentations possibles, c'est-à-dire les cas où on réalise l'opération avec une clé secrète secondaire avant, simultanément ou après l'opération avec les blocs de clé cryptographique secrète.

Etant donné que les opérations non linéaires sont les plus fragiles face aux attaques de type DPA ou SPA, elles sont protégées en priorité. Ainsi, lorsque l'algorithme cryptographique comprend des opérations linéaires après l'opération non linéaire 107, il est préférable de réaliser ces opérations uniquement sur le bloc de données en cours de cryptage 104 afin de limiter le nombre de calculs à exécuter.

La présente invention est décrite ci-après dans son application non limitative à un algorithme de type AES et plus particulièrement à un algorithme AES manipulant des clés de 16 octets.

La figure 2 illustre un procédé de cryptographie selon un algorithme de type AES. Un tel algorithme prend en entrée un bloc de données initial à crypter 201 pour fournir en sortie un bloc de données crypté correspondant 208.

L'algorithme comprend plusieurs tours (ou 'round' en anglais). Il est en

général basé sur une clé secrète principale K. Une clé principale peut avoir une taille de 128 bits, de 192 bits ou encore de 256 bits. Une telle clé est classiquement dérivée en une pluralité de clés, notées K_i. Les clés dérivées ont une taille de 16 octets pour un algorithme manipulant des clés de 128 bits, une taille de 24 octets pour un algorithme manipulant des clés de 192 bits et une taille de 32 octets pour un algorithme manipulant des clés de 256 bits.

On considère à titre d'exemple que la clé secrète principale K est de taille 128 bits et est dérivée en 10 clés de 16 octets, chacune de ces clés étant utilisées dans un tour spécifique.

Le message initial à crypter a une taille de 128 bits. On le traite généralement par blocs de données initiaux d'un octet 201. A un bloc de données d'un octet pour un tour déterminé de l'algorithme correspond un octet d'une clé.

On représente classiquement le message à crypter sous la forme d'une matrice d'état 4x4 de 16 blocs de données initiaux de 8 bits. Les blocs de données de 8 bits peuvent être traités les un après les autres ou encore simultanément. L'invention couvre toutes ces implémentations.

Ce message à crypter est en premier lieu transformé par une opération 202 à clé cryptographique secrète, classiquement référencée 'AddRoundKey'. Cette opération 202 ajoute au bloc de données initial 201 par un ou exclusif la clé principale K 203.

La clé secrète principale K 203 est utilisée lors de la première application de l'opération 202 pour obtenir un bloc de données en cours de cryptage. Puis le bloc de données entre dans un premier tour 204. Pour une clé de 128 bits, un tel algorithme comprend classiquement 9 tours 204 comprenant chacun les mêmes opérations successives suivantes :

- une opération 205, classiquement référencée 'ByteSub' ; cette dernière est un fonction non linéaire généralement implémentée sous forme d'une table de substitution ;
- une opération 206, classiquement référencée 'ShiftRow' ; cette dernière est une fonction opérant des décalages de lignes sur la matrice d'état;
- une opération 207, classiquement référencée 'MixColumn', cette dernière est une fonction de brouillage de colonnes sur la matrice d'état; et

- l'opération 202 'AddRoundKey' avec la clé K_r correspondante au tour T_r.

Puis, sur le bloc de données en cours de cryptage ainsi obtenu à l'issue des 9 tours, on applique à nouveau l'opération 205 'ByteSub', l'opération 206 'ShiftRow', et enfin l'opération 202 'AddRoundKey' avec la clé K_{10}.

Pour chacun des tours T_r, pour r égal 1 à 9, une clé secrète K_r dérivée de la clé secrète principale est utilisée pour l'exécution de l'opération 202 'AddRoudKey'.

On note $K_{i,r}$ la valeur de l'ième octet de la clé au tour T_r de l'AES, où i est compris entre 1 et L_r, où r est compris entre 1 et N_r avec N_r=10 et L_r=16 dans le cas où l'algorithme AES manipule des clés de 128 bits, N_r=12 et L_r=24 dans le cas où l'algorithme AES manipule des clés de 192 bits et N_r=16 et L_r=32 dans le cas où l'algorithme AES manipule des clés de 256 bits.

On note M le message d'entrée à crypter par l'algorithme et Mi, pour i égal 1 à 16, les blocs de données initiaux d'un octet correspondants. Ainsi, à chaque bloc de données d'un octet à traiter par l'algorithme, on applique chacun des blocs de clé cryptographique secrète d'un octet de la clé cryptographique secrète.

On réalise l'opération à clé secrète 202 'AddRoundKey' avec un bloc de clé cryptographique secrète pour obtenir un bloc de données en cours de cryptage et également avec au moins une clé secrète secondaire différente du bloc de clé cryptographique secrète pour obtenir un bloc de données secondaire. Plus le nombre de clés secrètes secondaire est important, plus la confidentialité de la clé cryptographique secrète est complexe et long à violer.

A cet effet, dans un mode de réalisation préféré, on construit de manière aléatoire une table initiale comprenant toutes les valeurs possibles d'un octet. Ainsi, une telle table comprend notamment le bloc de clé cryptographique secrète à appliquer au bloc de données par l'opération 'AddRoundKey' 202.

Une telle table de clés comprend 256 éléments, prenant les valeurs de 1 à 256. Ces valeurs sont rangées dans un ordre aléatoire.

Dans un mode de réalisation préférentiel, cette table est créée à chaque lancement de l'algorithme d'AES.

La figure 3 illustre une opération à clé secrète 102 selon un mode de

réalisation de l'invention. Une telle opération peut correspondre à l'opération 'AddRoundKey' 202 modifiée selon un mode de réalisation de l'invention.

L'opération 102 est une opération à réaliser avec un bloc de clé cryptographique secrète K 304 écrit sur n bits. Une table 301 comprend des éléments correspondant à toutes les valeurs possibles, soit 2^n éléments, rangés aléatoirement. A titre d'exemple, n est égal à 8. Un élément 304 correspond à un bloc de clé cryptographique secrète de l'opération à clé secrète 102. On recherche du bloc de clé cryptographique secrète dans la table 301 de préférence via une fonction résistante contre les attaques SPA. Ce type de fonction de recherche est bien connue de l'homme du métier et n'est pas détaillée dans ce document. De préférence, on stocke alors en mémoire l'adresse du bloc de clé cryptographique secrète.

On applique l'opération 102 à clé secrète au bloc de données 101 avec tous les éléments de la table 301 comprenant les valeurs de clés, soit simultanément, soit séquentiellement. On obtient alors une table 303 transformée qui comprend 2^n éléments, soit 256 éléments. Chacun de ces éléments correspond au résultat de l'opération 102 appliquée au bloc de données 101 avec la clé secrète située dans la table 301 à la même adresse que cet élément. Cette table 303 comprend notamment un élément 305 correspondant au bloc de données en cours de cryptage, ce bloc étant le résultat de l'opération à clé secrète 102 appliquée avec le bloc de clé cryptographique secrète 304.

L'opération étant appliquée de façon équiprobable avec toutes les valeurs de clé possible, cette étape est protégée contre toute attaque relative à une analyse des fuites d'information lors de l'exécution du calcul.

Puis, dans un algorithme de type AES, l'opération à clé secrète est suivie de l'opération non linéaire 107 'ByteSub'. Une telle opération peut être source d'informations précieuses lors d'attaques SPA ou DPA. Il est de ce fait très important de protéger son exécution. Ainsi, dans un mode de réalisation préféré de l'invention, on applique une telle opération sur tous les éléments de la table 303.

La figure 4 illustre l'exécution d'une opération non linéaire selon un mode de réalisation de la présente invention. Ainsi, l'opération non linéaire 107

est appliquée soit simultanément, soit séquentiellement, à tous les éléments de la table 303 pour fournir une table 402 qui comprend 2^n éléments, soit 256 éléments. Chaque élément correspond au résultat de l'opération non linéaire appliquée sur l'élément de la table 303 situé à la même adresse.

Ainsi, on est en mesure de récupérer le bloc de données en cours de cryptage 403 dès lors que l'on a stocké en mémoire l'adresse du bloc de clé cryptographique secrète dans la table 301 comprenant les clés.

On récupère alors le bloc de données en cours de cryptage 403 dans la table 402, sur la base de l'adresse du bloc de clé cryptographique secrète préalablement stockée en mémoire, de préférence, au moyen d'une fonction résistante contre les attaques SPA.

On peut alors réaliser les opérations 'ShiftRow' 206, 'MixColumn' 207 uniquement sur le bloc de données en cours de cryptage et non plus sur les blocs de données secondaires, issus d'opérations avec des clés secrètes secondaires et non des blocs de clé cryptographique secrète. Ces dernières opérations sont alors de préférence réalisées en appliquant des masques de valeurs aléatoires au bloc de données en cours de cryptage manipulé.

Dans un algorithme de type AES, toutes ou partie des opérations 202 'AddRoundKey' peuvent être réalisées selon un mode de réalisation de l'invention.

Dans certains cas, on peut souhaiter exécuter une partie seulement des opérations 'AddRoundKey' 202 et 'ByteSub' 205 de l'algorithme selon un mode de réalisation de la présente invention. Dans ce cas, on exécutera selon l'invention, de préférence les opérations 'AddRoundKey' 202 et 'ByteSub' 205 au début de l'algorithme, c'est-à-dire dans au moins le premier tour de l'algorithme, ou en fin d'algorithme, c'est-à-dire dans au moins le dernier tour de l'algorithme.

Afin d'améliorer la protection contre les attaques précédemment décrites, il est avantageux de masquer les blocs de données en cours de cryptage manipulés. Le masquage peut être réalisé aisément en ajoutant une valeur aléatoire par un ou exclusif.

La figure 5 illustre les étapes pour crypter un message de 16 octets selon un algorithme de type AES comprenant des calculs cryptographiques

exécutés selon un mode de réalisation de la présente invention, et plus particulièrement le passage au premier tour. Sur cette figure, la table contenant toutes les valeurs de clés est notée RAND[j]. Cette table comprend les valeurs de 1 à 256 aléatoirement rangées. Le message à crypter M est composé de 16 blocs de données d'un octet chacun, M_i pour i égal 1 à 16.

Ainsi, en début d'algorithme, on traite tout d'abord le premier octet M_1 501 du message à crypter M.

Dans l'étape 502, on réalise l'opération 'AddRoundKey' selon un mode de réalisation de l'invention. Ainsi, en une première étape 504, on recherche tout d'abord le bloc de clé cryptographique secrète $K_{i,1}$ dans la table RAND[j] via une fonction résistante contre les attaques SPA pour obtenir sa position dans cette table et on génère une valeur d'octet aléatoire A_i, utilisée pour masquer le bloc de données manipulé. Puis, on exécute une boucle pour j égal à 1 jusqu'à j égal 256, via les étapes 505, 506, 508, afin d'appliquer l'opération 'AddRoundKey' avec toutes les clés de la table RAND[j] suivi de l'exécution de l'opération 'ByteSub'. Lorsque tous les éléments de la table RAND[j] ont été traité, dans l'étape 511 on récupère, via une fonction résistante aux attaques SPA, le bloc de données en cours de cryptage correspondant au résultat des opérations sur le bloc de données M_1 avec le bloc de clé cryptographique secrète correspondant.

Ensuite, on incrémente i dans l'étape 512. On réitère ainsi toutes les opérations précédemment décrites sur tous les octets M_i du message à crypter. Puis, on applique sur les blocs de données en cours de cryptage ainsi obtenus les opérations 'SiftRows' et 'MixColumn', ces opérations étant préférablement réalisées de façon masquée.

La figure 6 illustre les étapes pour crypter un message de 16 octets selon un algorithme de type AES comprenant des calculs cryptographiques exécutés selon un mode de réalisation de la présente invention et plus particulièrement le passage entre deux tours consécutifs de l'algorithme.

L'étape 602 représente l'opération 'AddRoundKey' exécutée en fin d'un tour de l'algorithme. Les étapes 602 et 603 sont similaires aux étapes de la figure 5 précédemment décrite. On note que dans l'étape 606, on masque le calcul en ajoutant par un ou exclusif un masque B, qui est une valeur aléatoire.

Ainsi, lors d'une attaque sur l'opération non linéaire, on collecte de manière équiprobable l'ensemble des fuites d'information liées à l'opération non linéaire de substitution puisque cette dernière est réalisée sur tous les blocs de données secondaires et le bloc de données en cours de cryptage. De cette manière, lors d'une attaque DPA menée lors de l'exécution de calculs de l'algorithme selon un mode de réalisation de l'invention, on peut détecter 256 biais, 1 biais pour chaque octet de clé. En conséquence toutes les hypothèses de clé sont validées par une attaque de ce type. La confidentialité des clés secrètes est ainsi préservée.

Afin de préserver une bonne performance d'exécution de l'algorithme cryptographique, on peut avantageusement réaliser simultanément et donc parallèlement une partie des calculs exécutés selon l'invention.

REVENDICATIONS

1. Procédé d'exécution d'un calcul cryptographique dans un composant électronique, selon un algorithme cryptographique déterminé incluant au moins une opération à clé secrète (102) à réaliser sur un bloc de données (101) avec une clé cryptographique secrète (103) comprenant m blocs de clé cryptographique secrète de n bits et une opération non linéaire (107), ledit procédé comprenant, pour un bloc de clé cryptographique secrète donné, les étapes suivantes consistant à :

- déterminer au moins une clé secrète secondaire (105) sur n bits différente dudit bloc de clé cryptographique secrète ;
- réaliser ladite opération à clé secrète (102) avec ledit bloc de clé cryptographique secrète (103) et réaliser ladite opération à clé secrète (102) avec ladite clé secrète secondaire sur un bloc de données (101) et obtenir respectivement un bloc de données en cours de cryptage (104) et un bloc de données secondaire (106);
- réaliser ladite opération non linéaire (107) sur ledit bloc de données en cours de cryptage (104) et sur ledit bloc de données secondaire (106) ;
- fournir un bloc de données crypté (108) à partir du bloc de données en cours de cryptage.

2. Procédé selon la revendication 1, comprenant, pour un bloc de clé cryptographique secrète, les étapes consistant à:

/a/ déterminer et arranger aléatoirement un nombre déterminé p de clés secrètes dans une table initiale comprenant ledit bloc de clé cryptographique secrète;

/b/ stocker en mémoire l'adresse correspondant audit bloc de clé cryptographique secrète dans la table initiale;

/c/ appliquer l'opération à clé secrète au bloc de données (101) avec les clés de la table initiale (301) et obtenir une première table transformée (303) de p premiers éléments, chaque premier élément correspondant au résultat de l'opération à clé secrète (102) appliquée au bloc de données (101) avec la clé

située dans la table initiale à la même adresse que ledit premier élément;

/d/ appliquer ladite opération non linéaire (107) aux éléments de ladite première table transformée (303) et obtenir une seconde table transformée (402) de p seconds éléments, chaque second élément correspondant au résultat de l'opération non linéaire (107) appliquée au premier élément situé à la même adresse dans la première table transformé (303) que ledit second élément;

/e/ récupérer, dans la seconde table transformée (402), l'élément correspondant au bloc de données en cours de cryptage (403) situé à l'adresse dudit bloc de clé cryptographique secrète.

3. Procédé selon la revendication 2, suivant lequel le nombre p de clé secrètes dans la table initiale est égal à 2^n.

4. Procédé selon la revendication 2 ou 3, suivant lequel on récupère l'élément correspondant au bloc de données en cours de cryptage (403), dans la seconde table transformée, via une fonction résistante contre une attaque de type SPA pour 'Simple Power Analysis' prenant en paramètre l'adresse de la clé cryptographique secrète

5. Procédé selon l'une quelconque des revendications 2 à 4, suivant lequel l'algorithme cryptographique comprend un nombre déterminé de tours, chacun incluant au moins une opération à clé cryptographique secrète précédant une opération non linéaire;
et suivant lequel on réalise les étapes /a/ à /e/ pour au moins le premier tour et au moins le dernier tour de l'algorithme cryptographique.

6. Procédé selon l'une quelconque des revendications 2 à 5, suivant lequel on réalise l'étape d'arrangement aléatoire au début de l'algorithme cryptographique.

7. Procédé selon l'une quelconque des revendications précédentes, suivant lequel on réalise simultanément l'opération à clé secrète (102) avec un des

blocs de clé cryptographique secrète et avec la clé secrète secondaire et/ou on réalise simultanément l'opération non linéaire sur le bloc de données et les blocs de données secondaires.

8. Procédé selon l'une quelconque des revendications précédentes, suivant lequel l'algorithme cryptographique est l'AES.

9. Procédé selon l'une quelconque des revendications précédentes, suivant lequel l'une au moins des opérations de l'algorithme cryptographique est réalisée sur le bloc de données en cours de cryptage masqué avec une valeur aléatoire.

10. Composant électronique adapté pour exécuter un calcul cryptographique selon un algorithme cryptographique déterminé incluant au moins une opération à clé secrète (102) à réaliser avec une clé cryptographique secrète (103) comprenant m blocs de clé cryptographique secrète de n bits sur un bloc de données (101) et une opération non linéaire (107), comprenant des moyens agencés pour mettre en œuvre un procédé selon l'une quelconque des revendications précédentes.

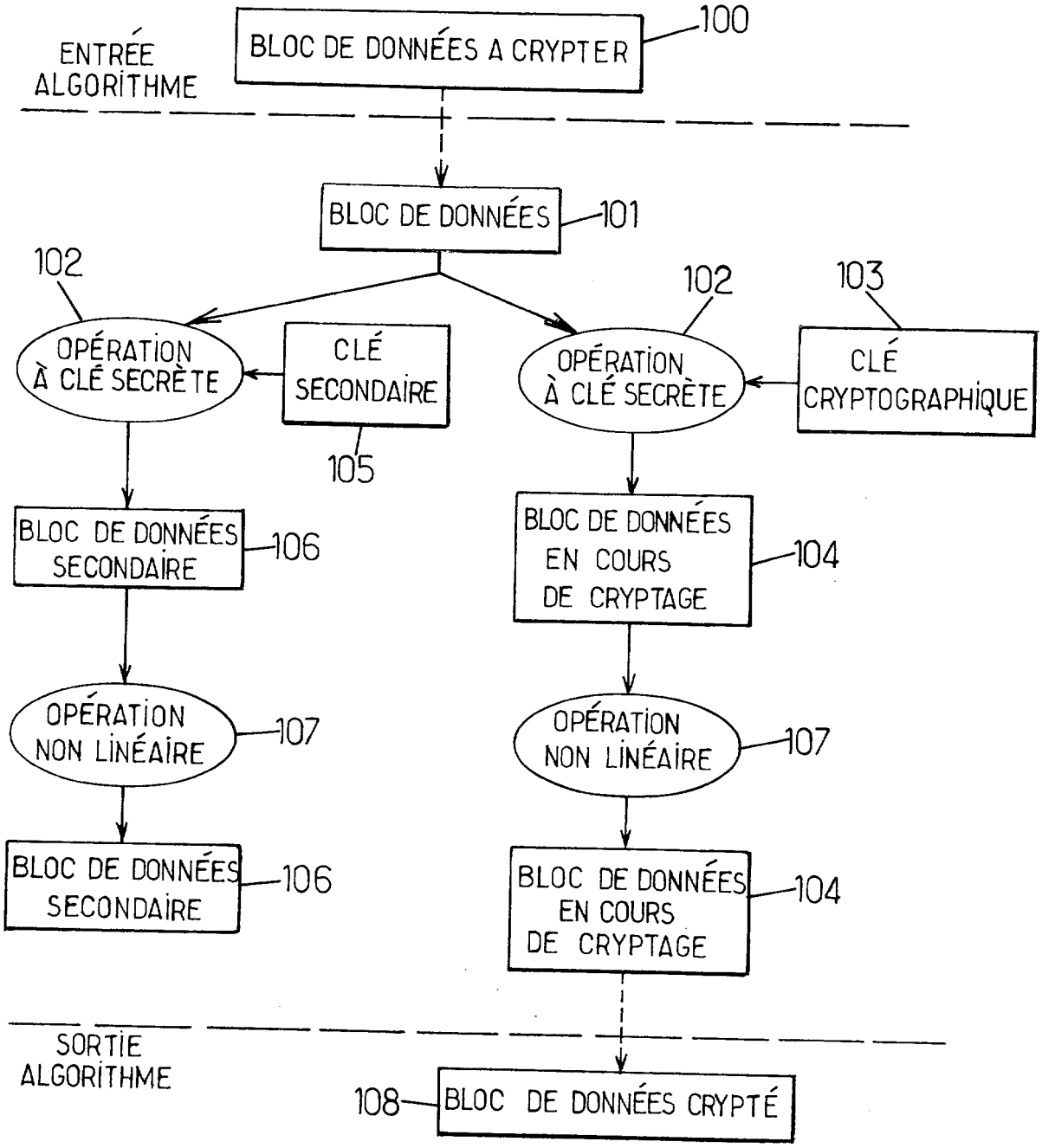

ENTRÉE ALGORITHME
BLOC DE DONNÉES A CRYPTER
100
BLOC DE DONNÉES
101
102
OPÉRATION À CLÉ SECRÈTE
CLÉ SECONDAIRE
105
102
OPÉRATION À CLÉ SECRÈTE
103
CLÉ CRYPTOGRAPHIQUE
BLOC DE DONNÉES SECONDAIRE
106
BLOC DE DONNÉES EN COURS DE CRYPTAGE
104
OPÉRATION NON LINÉAIRE
107
OPÉRATION NON LINÉAIRE
107
BLOC DE DONNÉES SECONDAIRE
106
BLOC DE DONNÉES EN COURS DE CRYPTAGE
104
SORTIE ALGORITHME
108
BLOC DE DONNÉES CRYPTÉ

FIG.1.

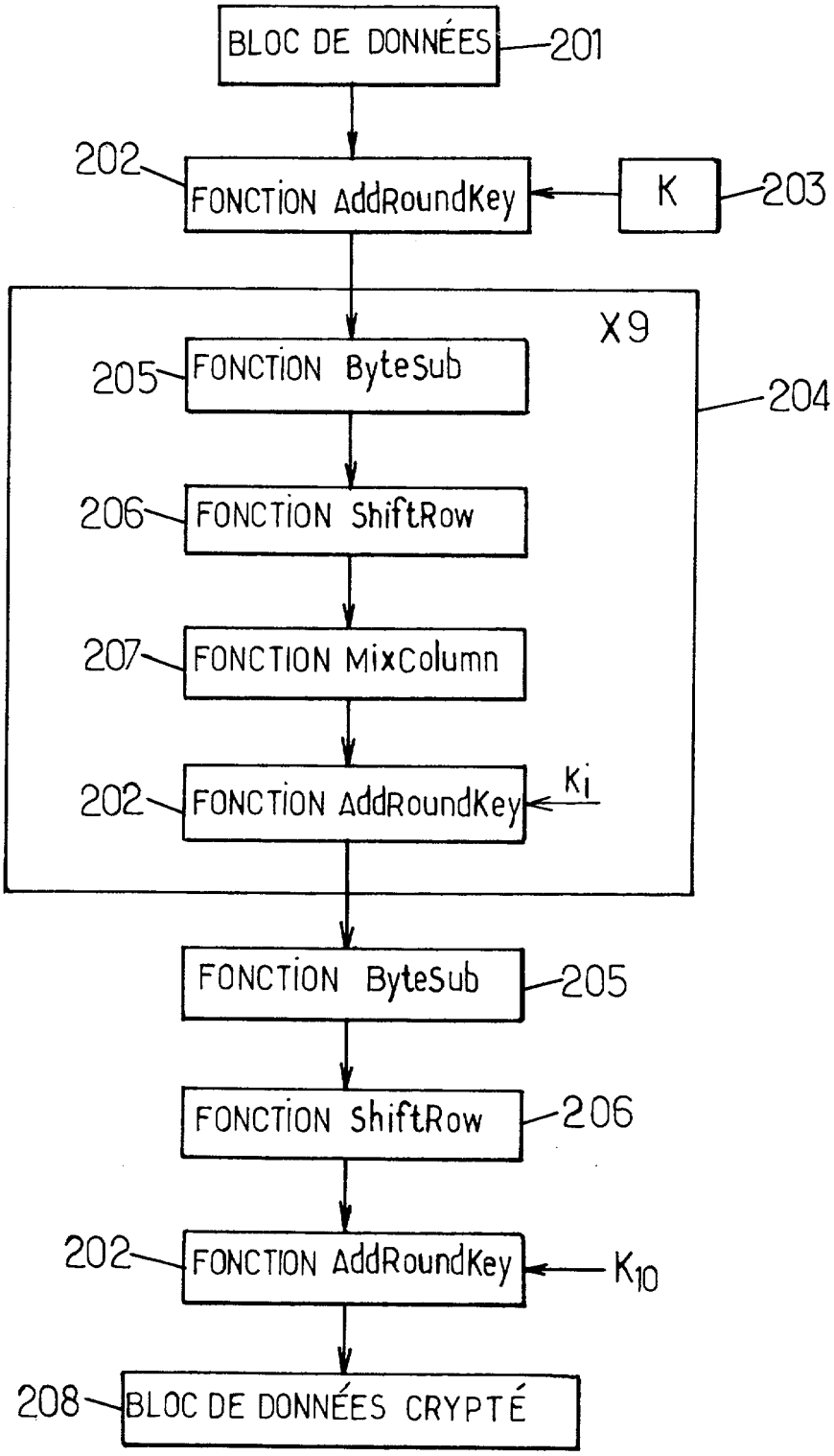

BLOC DE DONNÉES
201
202
FONCTION AddRoundKey
K
203
X9
205
FONCTION ByteSub
204
206
FONCTION ShiftRow
207
FONCTION MixColumn
202
FONCTION AddRoundKey
Ki
FONCTION ByteSub
205
FONCTION ShiftRow
206
202
FONCTION AddRoundKey
K10
208
BLOC DE DONNÉES CRYPTÉ

FIG.2.

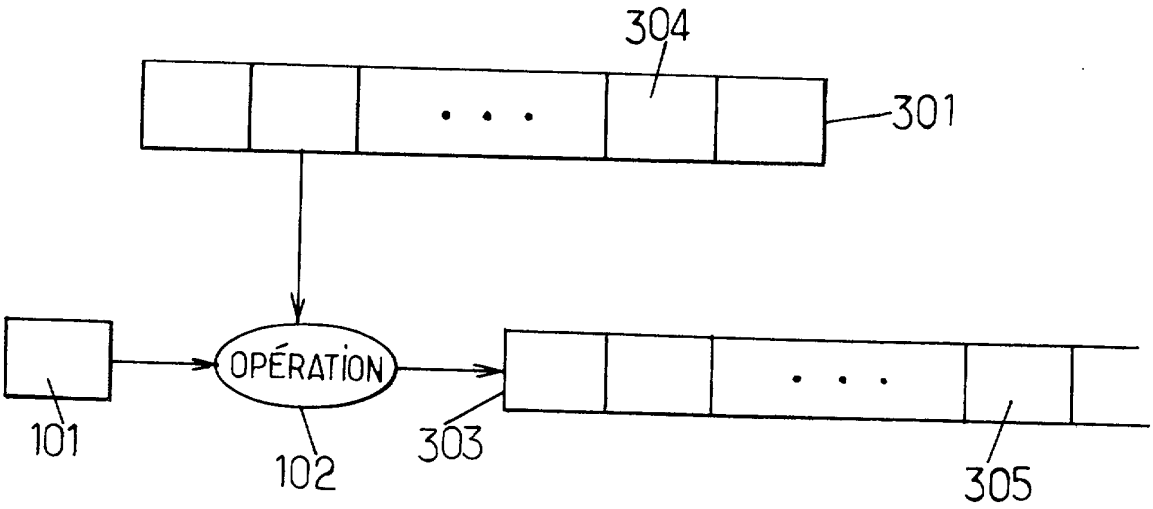

304
301
OPÉRATION
101
102
303
305

FIG.3.

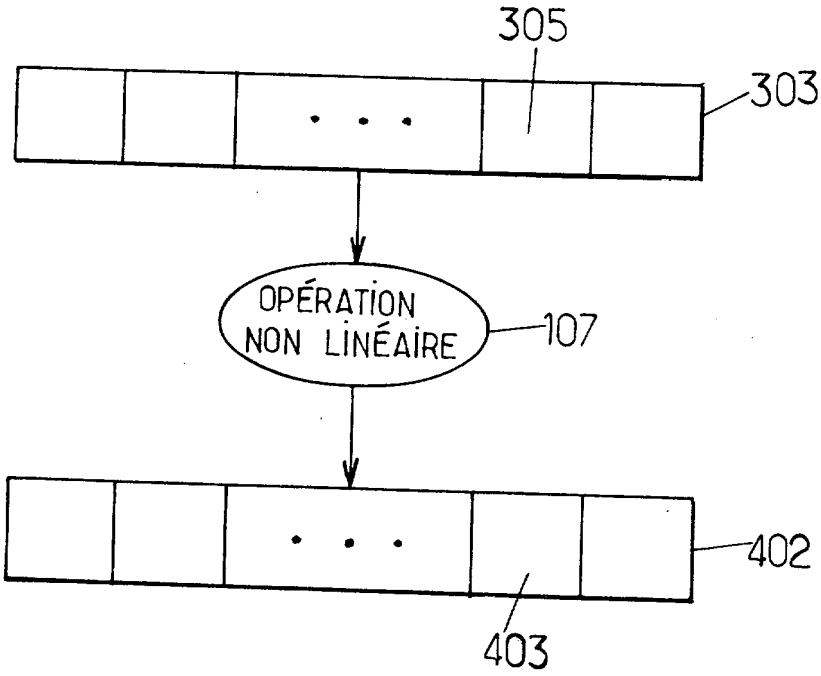

305
303
OPÉRATION
NON LINÉAIRE
107
402
403

FIG.4.

FIG.5.

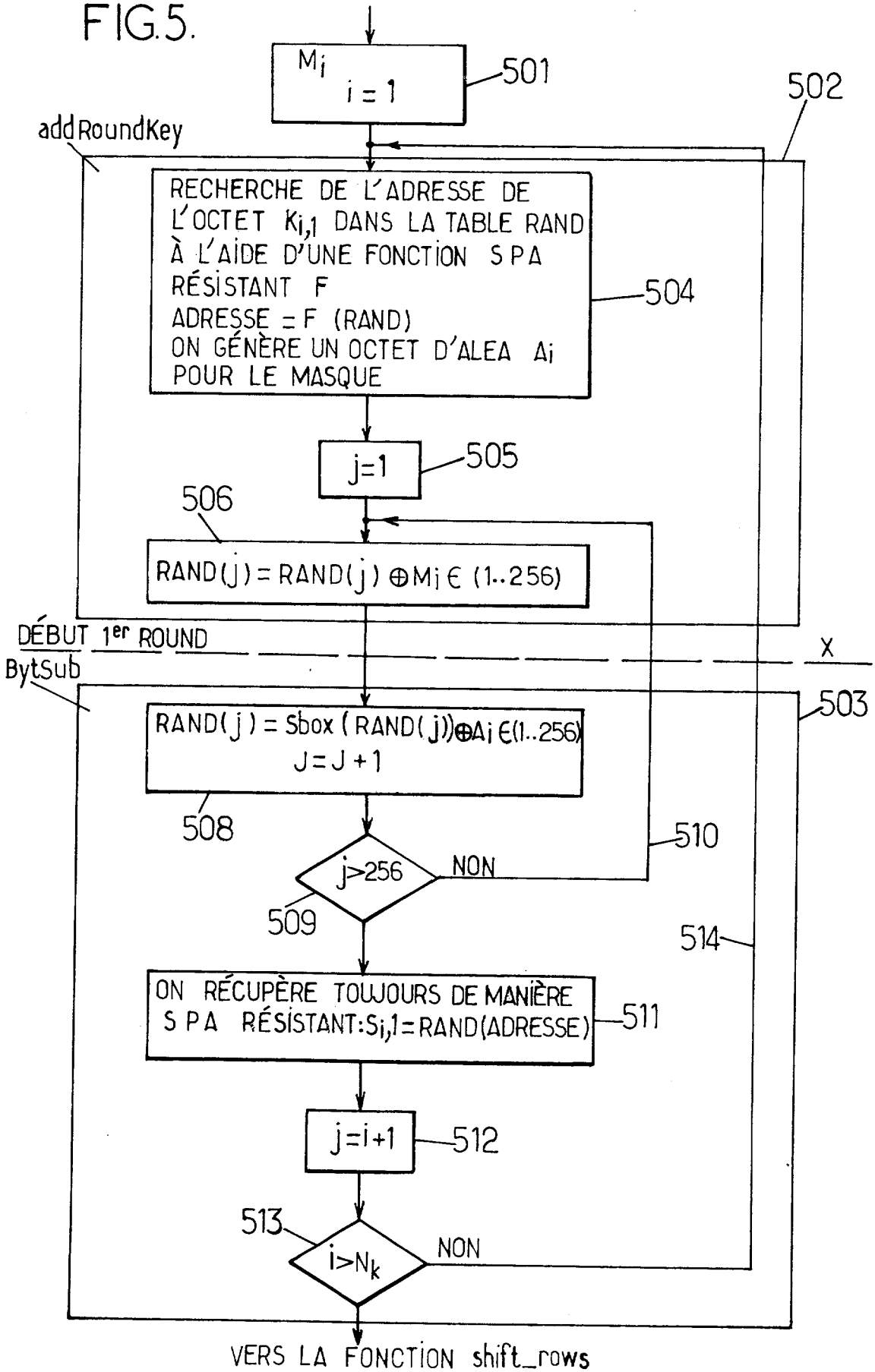
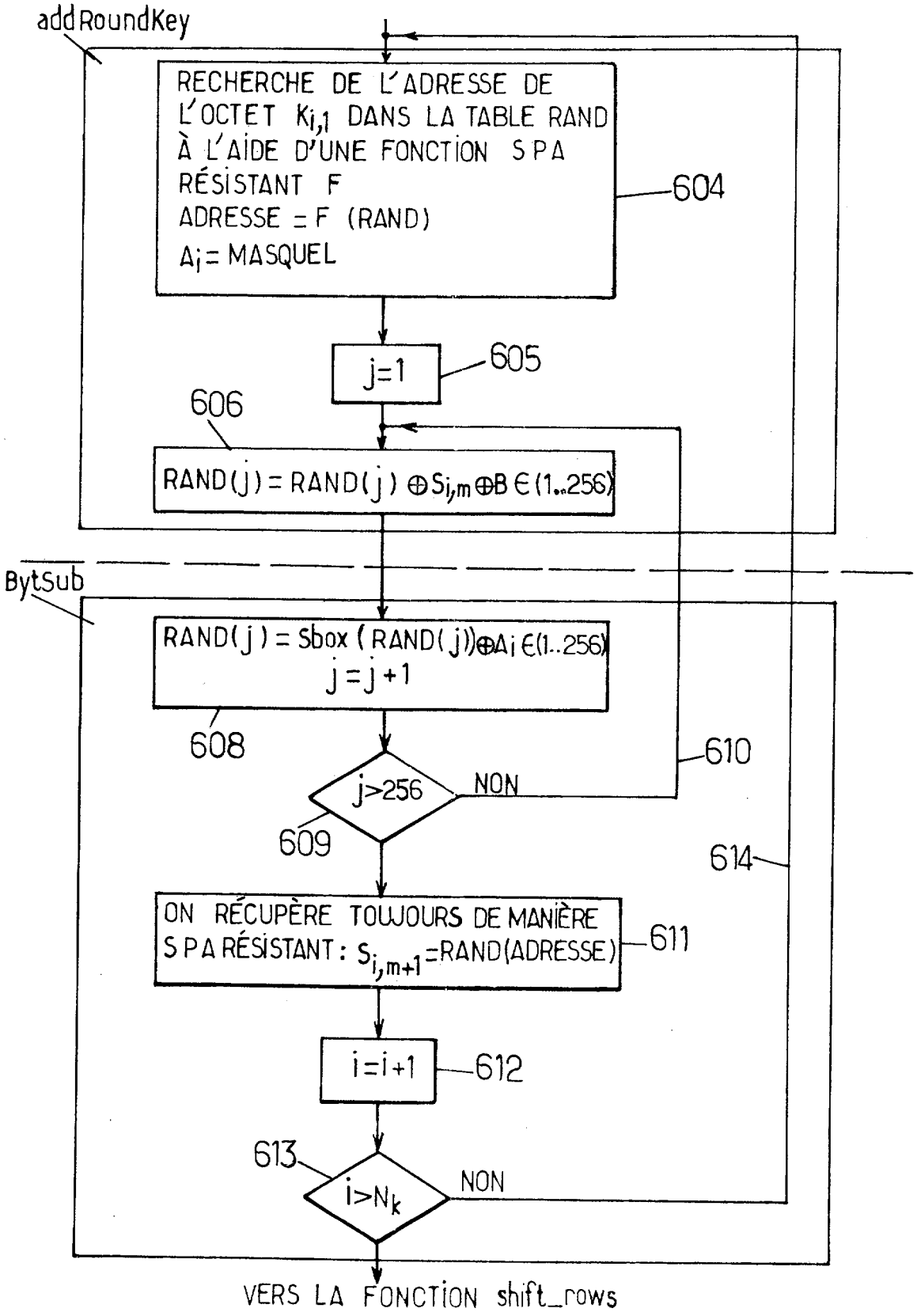

Mi
i = 1
501
502
addRoundKey
RECHERCHE DE L'ADRESSE DE L'OCTET Ki,1 DANS LA TABLE RAND À L'AIDE D'UNE FONCTION S PA RÉSISTANT F
ADRESSE = F (RAND)
ON GÉNÈRE UN OCTET D'ALEA Ai POUR LE MASQUE
504
j=1
505
506
RAND(j) = RAND(j) ⊕ Mj ∈ (1..256)
DÉBUT 1er ROUND
X
BytSub
503
RAND(j) = Sbox (RAND(j)) ⊕ Ai ∈(1..256)
J = J + 1
508
510
j>256
NON
509
514
ON RÉCUPÈRE TOUJOURS DE MANIÈRE S PA RÉSISTANT: Si,1 = RAND(ADRESSE)
511
j=i+1
512
513
i>Nk
NON
VERS LA FONCTION shift_rows

FIG.6.


addRoundKey
RECHERCHE DE L'ADRESSE DE L'OCTET $K_{i,1}$ DANS LA TABLE RAND À L'AIDE D'UNE FONCTION SPA RÉSISTANT F
ADRESSE = F (RAND)
A_i = MASQUEL
604
j=1
605
606
RAND(j) = RAND(j) ⊕ $S_{i,m}$ ⊕B ∈ (1..256)
BytSub
RAND(j) = Sbox (RAND(j)) ⊕A_i ∈(1..256)
j = j + 1
608
610
j>256
NON
609
614
ON RÉCUPÈRE TOUJOURS DE MANIÈRE SPA RÉSISTANT : $S_{i,m+1}$ =RAND(ADRESSE)
611
i=i+1
612
613
$i>N_k$
NON
VERS LA FONCTION shift_rows

RÉPUBLIQUE FRANÇAISE

2873523

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement national

FA 654180
FR 0408139

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		
X	DE 102 23 175 A (INFINEON TECHNOLOGIES AG) 11 décembre 2003 (2003-12-11) * alinéas [0013], [0014], [0016] - [0018], [0020], [0025], [0027] - [0030] *	1-10	H04L9/30 H04L9/06
X	FR 2 831 739 A (GEMPLUS CARD INT) 2 mai 2003 (2003-05-02) * page 4, ligne 24 - ligne 31 * * page 5, ligne 18 - ligne 26 * * page 7, ligne 8 - page 8, ligne 23 *	1-10	
X	GB 2 345 229 A (MOTOROLA LTD) 28 juin 2000 (2000-06-28)	1,10	
A	* page 11, ligne 10 - ligne 28 *	2-9	
A	EP 1 263 163 A (SAGEM) 4 décembre 2002 (2002-12-04) * alinéas [0018], [0019]; figure 1b *	1-10	

DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)

H04L

Date d'achèvement de la recherche: 30 décembre 2004

Examinateur: Cretaine, P

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 0408139 FA 654180

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 30-12-2004
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
DE 10223175 A	11-12-2003	DE 10223175 A1	11-12-2003
FR 2831739 A	02-05-2003	FR 2831739 A1	02-05-2003
		EP 1442556 A2	04-08-2004
		WO 03039065 A2	08-05-2003
GB 2345229 A	28-06-2000	AUCUN	
EP 1263163 A	04-12-2002	FR 2825542 A1	06-12-2002
		EP 1263163 A1	04-12-2002

EPO FORM P0465