

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
18 juin 2009 (18.06.2009)

PCT

(10) Numéro de publication internationale
WO 2009/074728 A1

(51) Classification internationale des brevets :
H04L 9/06 (2006.01)

(FR). **PROUFF, Emmanuel** [FR/FR]; 79, rue Lamarck,
F-75018 Paris (FR).

(21) Numéro de la demande internationale :
PCT/FR2007/002063

(74) **Mandataire : SANTARELLI**; 14, avenue de la Grande-
Armée, B.P. 237, F-75822 Paris Cedex 17 (FR).

(22) Date de dépôt international :
13 décembre 2007 (13.12.2007)

(81) **États désignés** (sauf indication contraire, pour tout titre de
protection nationale disponible) : AE, AG, AL, AM, AT,
AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES,
FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN,
IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR,
LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX,
MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO,
RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(25) Langue de dépôt : français

(26) Langue de publication : français

(71) **Déposant** (pour tous les États désignés sauf US) :
OBERTHUR TECHNOLOGIES [FR/FR]; 102, boule-
vard Malesherbes, F-75017 Paris (FR).

(72) **Inventeurs; et**

(75) **Inventeurs/Déposants** (pour US seulement) : **RIVAIN,**
Matthieu [FR/FR]; 35, rue Guy Moquet, F-75017 Paris

(84) **États désignés** (sauf indication contraire, pour tout titre
de protection régionale disponible) : ARIPO (BW, GH,

[Suite sur la page suivante]

(54) **Title:** METHOD FOR CRYPTOGRAPHIC DATA PROCESSING, PARTICULARLY USING AN S BOX, AND RELATED
DEVICE AND SOFTWARE

(54) **Titre :** PROCEDE DE TRAITEMENT CRYPTOGRAPHIQUE DE DONNEES, NOTAMMENT A L'AIDE D'UNE BOITE
S, DISPOSITIF ET PROGRAMME ASSOCIES

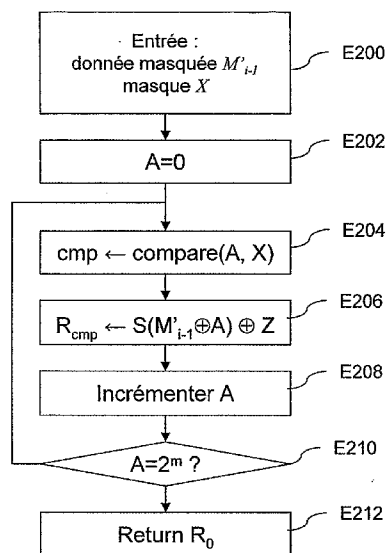


Figure 4

E200 Input: masked datum M'_{i-1} , mask X

E208 Increment A

(57) **Abstract:** The invention relates to a method and an associated system for the cryptographic processing of data in a digital format, wherein the method is implemented by an electronic entity (10) and comprises the conversion (E110) of an input datum (M'_{i-1}), masked by at least a first input mask (X, X_1, X_2), into an output datum (M'_i), said conversion using a conversion table (S) representing a non-linear function (F), and said method being characterised by the following steps: a- comparison (E204, E304, E406) between a masking value (A) and said first input mask (X, X_1, X_2); b- determining (E206, E306, E408) using said conversion table an output datum corresponding to the masked input datum to which said masking value is applied; c- storing (E206, E306, E408) the determined output datum in a register (R_0, R_1) depending on the result of said comparison.

(57) **Abrégé :** L'invention concerne un procédé et un système associé de traitement cryptographique de données représentées sous forme numérique, le procédé étant mise en œuvre par une entité électronique (10) et comprenant une transformation (E110) d'une donnée d'entrée (M'_{i-1}), masquée par au moins un premier masque d'entrée (X, X_1, X_2), en une donnée de sortie (M'_i), ladite transformation mettant en œuvre une table de conversion (S) représentant une fonction non linéaire (F), le procédé étant caractérisé par les étapes suivantes: a- comparaison (E204, E304, E406) entre une valeur de masquage (A) et ledit premier masque d'entrée (X, X_1, X_2); b- détermination (E206, E306, E408), à l'aide de ladite table de conversion, d'une donnée de sortie correspondant à la donnée d'entrée masquée à laquelle on applique ladite valeur de masquage; c- stockage (E206, E306, E408), dans un registre (R_0, R_1) dépendant du résultat de ladite comparaison, de la donnée de sortie déterminée.

WO 2009/074728 A1



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

— *relative à des divulgations non opposables ou à des exceptions au défaut de nouveauté (règle 4.17.v))*

Publiée :

— *avec rapport de recherche internationale*

Déclarations en vertu de la règle 4.17 :

— *relative à la qualité d'inventeur (règle 4.17.iv))*

PROCEDE DE TRAITEMENT CRYPTOGRAPHIQUE DE DONNEES,
NOTAMMENT A L'AIDE D'UNE BOITE S, DISPOSITIF ET PROGRAMME
ASSOCIES

5 L'invention concerne un procédé de traitement cryptographique de données, ainsi qu'un dispositif et un programme associés.

 Dans de tels procédés, on a fréquemment recours au masquage des données afin de lutter contre les attaques, par exemple du type analyse de courant (notamment les attaques de type DPA ainsi dénommées d'après
10 l'anglais "*Differential Power Analysis*") ou du type analyse de rayonnement électromagnétique.

 Les techniques de masquage consistent à combiner la donnée (c'est-à-dire en pratique le nombre) que l'on souhaite utiliser (en pratique, auquel on souhaite faire subir une opération) avec un nombre imprévisible pour
15 un attaquant extérieur (en général un nombre aléatoire ou pseudo-aléatoire) ; ainsi, les valeurs impliquées sont chaque fois différentes même en utilisant une donnée constante en entrée, ce qui rend impossible pour l'attaquant de déduire les données internes du procédé (et notamment les clés cryptographiques qu'il utilise) à partir de mesures faites de l'extérieur.

20 Une part de la sécurité cryptographique est obtenue par l'utilisation de fonctions non-linéaires. Par exemple, il est courant de modéliser un chiffrement par bloc (*block cipher*) par la combinaison de fonctions affines et de fonctions non-linéaires. Les réalisations de telles fonctions non-linéaires sont particulièrement difficiles à protéger par masquage, du fait de la non-linéarité
25 par rapport à l'opération de masquage.

 Un exemple de procédé de traitement cryptographique est décrit par la demande de brevet internationale WO 2007/116140 qui met en œuvre une fonction non-linéaire de type boîte-S (S-Box selon la terminologie anglo-saxonne spécialisée du domaine considéré) appliquée à ces données
30 masquées.

 De telles boîtes-S ou "*S-Boxes*" sont réalisées en pratique au moyen d'une table de correspondance (fréquemment dénommée table-S ou look-up

table [LUT] dans ce même domaine) mémorisée dans le dispositif cryptographique.

Les améliorations apportées par la demande de brevet susvisée permettent désormais d'utiliser une même table-S alors que de nouveaux
5 masques indépendants sont utilisés lors de nouveaux calculs cryptographiques impliquant la boîte-S. On évite ainsi le re-calcul de tables-S.

Ainsi, lorsque l'on souhaite calculer une donnée de sortie de la boîte-S à partir d'une donnée d'entrée, il est nécessaire d'accéder à la table-S, par exemple pour y lire la donnée de sortie correspondant à celle d'entrée.

10 L'accès répété, lors d'un processus cryptographique, à la même entrée (indice) de la table-S est toutefois préjudiciable en ce qu'une analyse de courant permet d'en détecter la valeur et donc la donnée d'entrée.

La demande susvisée a ainsi proposé des solutions alternatives par lesquelles le calcul de la donnée de sortie de la boîte-S est effectué par une
15 combinaison linéaire portant sur les données de sortie correspondant aux autres données d'entrée.

Néanmoins, l'accès à toutes les données de sortie génère des difficultés quant aux accès mémoire, stockage des données et émission de signaux électromagnétiques accessibles aux attaquants.

20 Afin de réponse à ces problèmes, et notamment d'effectuer un calcul sur une boîte-S sans surcoût et en toute sécurité, l'invention prévoit un procédé de traitement cryptographique de données représentées sous forme numérique, le procédé étant mise en œuvre par une entité électronique et comprenant une transformation d'une donnée d'entrée (par exemple une donnée sensible telle
25 qu'une clé cryptographique), masquée par au moins un premier masque d'entrée, en une donnée de sortie, ladite transformation mettant en œuvre une table de conversion (par exemple stockée en mémoire de ladite entité électronique) représentant une fonction non linéaire, le procédé étant caractérisé par les étapes suivantes:

30 a- comparaison entre une valeur de masquage et ledit premier masque d'entrée;

b- détermination, à l'aide de ladite table de conversion, d'une donnée de sortie correspondant à la donnée d'entrée masquée à laquelle on applique ladite valeur de masquage;

5 c- stockage, dans un registre dépendant du résultat de ladite comparaison, de la donnée de sortie déterminée.

La détermination met en œuvre les boîtes-S. Une lecture de cette boîte-S est effectuée pour l'entrée formée de la donnée d'entrée masquée à laquelle on applique la valeur de masquage.

10 Selon l'invention, on utilise donc des registres qui sont, par nature, d'accès plus simple que les mémoires.

Le registre associé à une comparaison positive contient alors la donnée de sortie souhaitée (correspondant à la donnée d'entrée non masquée), alors que l'autre registre comprend l'enregistrement de données de sortie associées aux entrées différentes de la donnée d'entrée considérée. Ainsi, il est possible de mettre en place des opérations factices améliorant la sécurité, qui n'impactent pas le registre associé à la donnée de sortie souhaitée.

Cette table-S peut alors être stockée en mémoire non volatile sans occuper la mémoire vive, parfois réduite, allouées au microcontrôleur.

20 L'entité électronique est apte à manipuler les données numériques traitées, par exemple au moyen d'un microprocesseur.

Le procédé est, par exemple, un procédé de chiffage ou de déchiffage de données numériques, représentant en général un message, mais également potentiellement une clé cryptographique, une donnée intermédiaire, ou une partie seulement de tels éléments.

25 On note que l'application de la valeur de masquage à la donnée d'entrée masquée est généralement une opération de suppression de masque (opération de démasquage) à partir de ladite valeur de masquage. En particulier, cette opération de démasquage peut consister en une nouvelle opération de masquage par la valeur de masquage, auquel cas les fonctions de masquage sont généralement des fonctions involutives (la double-composition de la fonction équivaut à la fonction identité), comme par exemple la fonction binaire OU EXCLUSIF (XOR).

Selon un mode de réalisation, on itère les étapes *a*, *b* et *c* pour une pluralité de valeurs de masquage.

Ainsi, en accédant plusieurs fois à la table de conversion table-S, on ne divulgue pas l'entrée de la table associée à la donnée d'entrée; on récupère
5 toutefois aisément dans le registre associé à la comparaison positive, la donnée de sortie.

Ainsi, la donnée de sortie ne découle pas d'un calcul complexe sur les autres données de sortie, mais de la lecture de sa valeur dans la table de conversion table-S et son stockage temporaire dans un registre, sans recalcul
10 de table.

Notamment, on peut itérer les étapes *a*, *b* et *c* pour un nombre *N* de valeurs de masquage égale à la taille de la table de conversion, c'est-à-dire le nombre de données de sortie stockées dans ladite table.

En particulier, lesdites *N* valeurs de masquage sont choisies de sorte
15 qu'appliquées à la donnée d'entrée masquée, elles génèrent l'ensemble des valeurs d'entrée de ladite table de conversion, et donc l'ensemble des valeurs de sortie possibles.

Ainsi, au travers des *N* itérations on accède à chacune des entrées de ladite table de conversion. Comme aucun accès n'est privilégié, la sécurité
20 du système/procédé est renforcée.

En pratique on utilisera des masques binaires de même longueur que la donnée d'entrée binaire. Ainsi l'ensemble des masques binaires correspond à l'ensemble des entrées de la boîte-S (table-S).

Les masques binaires associés à une fonction d'addition booléenne
25 ont l'avantage de constituer une fonction de masquage involutive simple, permettant de retrouver la valeur de donnée d'entrée lorsque celle-ci est masquée deux fois par le même masque.

L'invention s'applique ainsi à tout ordre de masquage, c'est-à-dire quelque soit le nombre de masques aléatoires appliquée à la donnée.

30 Notamment, dans un mode de réalisation correspondant à l'ordre 2, ladite donnée d'entrée est masquée par un premier et un deuxième masques d'entrée, ladite étape de comparaison étant alors réalisée entre ledit premier

masque d'entrée et une combinaison entre ladite valeur de masquage et ledit deuxième masque d'entrée.

Ainsi, la comparaison est positive lorsque la valeur de masquage correspond à la combinaison, par exemple addition booléenne, des deux masques d'entrée. Par conséquent, la donnée contenue dans le registre associée est celle de sortie correspondant à l'entrée masquée par les deux masques d'entrée puis masquée à nouveau par la valeur de masquage ci-dessus. La fonction de masquage étant involutive, la donnée de sortie correspond alors à la donnée d'entrée non masquée.

10 En pratique, le calcul peut être obtenu en réalisant pour $A=0$ à 2^m-1 ,

$$cmp \leftarrow compare(X_1 \oplus A, X_2)$$

$$R_{cmp} \leftarrow S(M' \oplus A) \oplus Z_1 \oplus Z_2$$

return R_0

15 où X_1 et X_2 sont deux masques d'entrée, m la dimension d'entrée de la table-S, M' la donnée masquée, R_0 et R_1 deux registres, Z_1 et Z_2 deux masques de sortie et *compare* une fonction retournant 0 en cas d'égalité, 1 sinon.

En particulier, la fonction de masquage peut être commutative, ce qui permet d'appréhender le double (voire plus) masquage quelque soit l'ordre d'application des masques d'entrée ou de sortie comme mentionné plus loin.

20 Dans un mode de réalisation, ledit stockage à l'étape c comprend l'addition de la donnée déterminée à la valeur dudit registre. En particulier, une pluralité d'itérations des étapes a, b et c est prévue et le procédé comprend une étape, ultérieure aux itérations, consistant à retourner une valeur de sortie formée à partir desdits registres.

En pratique, le calcul peut être obtenu en réalisant pour $A=0$ à 2^m-1 ,

$$cmp \leftarrow compare(A, X)$$

$$R_{cmp} \leftarrow R_{cmp} \oplus S(M' \oplus A)$$

$$cmp \leftarrow compare(R_0, R_1)$$

30 *return* $R_0 \oplus (cmp \times R_1)$

où X est le masque d'entrée, m la dimension d'entrée de la table-S, M' la donnée masquée, R_0 et R_1 deux registres et *compare* une fonction retournant 0 en cas d'égalité, 1 sinon.

5 Pour protéger la donnée de sortie obtenue par le mécanisme de masquage évoqué plus haut, on prévoit que les (deux) registres sont initialisés (avant de commencer les itérations) à une valeur de masquage Z .

Dans un mode de réalisation pratique, la comparaison met en œuvre une fonction de deux paramètres dont le résultat est un booléen (b), par exemple 0, en cas d'égalité entre les deux paramètres et le complémentaire du booléen ($\bar{b}$), par exemple 1, sinon, lesdits registres étant indexés par ledit résultat de la fonction de comparaison.

$$\text{En pratique: } compare(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

En particulier, le procédé comprend une étape préalable (auxdites itérations et non à chaque itération) de génération pseudo-aléatoire dudit booléen.

En pratique: $b \leftarrow \text{rand}(1)$

$$\text{puis } compare_b(x, y) = \begin{cases} b & \text{if } x = y \\ \bar{b} & \text{if } x \neq y \end{cases}$$

En particulier, la comparaison comprend la lecture de l'entrée d'une table T à un index fonction des deux paramètres et d'une valeur prédéterminée.

20 En pratique, on prévoit d'initialiser chaque entrée de la table T avec la valeur du complémentaire du booléen, de générer pseudo-aléatoirement un index et d'enregistrer ledit booléen à l'entrée de la table T correspondant audit index:

$r3 \leftarrow \text{rand}(m)$, $r3$ étant l'index généré aléatoirement

25 $T \leftarrow \{\bar{b}; \bar{b}; \bar{b}; \dots; \bar{b}\}$, b étant le booléen généré

$T[r3] \leftarrow b$

Puis d'effectuer la comparaison en lisant l'entrée $T[(x \oplus r3) \oplus y]$ où x est la valeur de masquage (éventuellement combinée avec un deuxième masque) de l'itération et y le premier masque.

Selon un mode de réalisation particulièrement intéressant, lesdits masques peuvent être des masques additifs, par exemple des masques booléens. Ainsi, on peut mettre en œuvre aisément l'addition booléenne à l'aide de fonctions OU EXCLUSIF (XOR).

5 Selon une caractéristique particulière de l'invention, le procédé comprend, avec l'étape c, le masquage de ladite donnée de sortie par au moins un premier masque de sortie.

En particulier, ce masquage est double avec un deuxième masque de sortie dans le cas d'un masquage d'ordre 2, afin de sécuriser le système et
10 le procédé selon l'invention contre des attaques d'ordre 2.

Lorsqu'un simple masque est utilisé pour masquer une donnée (masquage d'ordre 1), la fuite d'information sur la donnée masquée peut-être ciblée conjointement avec la fuite d'information sur le masque afin de retrouver de l'information sur la donnée en claire. Ce type d'attaques couramment
15 appelées attaques d'ordre 2 peut être évité par l'utilisation d'un second masque (masquage d'ordre 2).

Selon une caractéristique pratique, lesdits registres sont des registres internes et donc mieux protégés, notamment d'un microcontrôleur sécurisé, par exemple celui mettant en œuvre ledit procédé. On limite ainsi le
20 transfert de données au travers d'un bus de données moins sécurisé.

Le procédé est par exemple mis en œuvre par une séquence d'instructions mémorisées dans l'entité électronique et exécutées par un microprocesseur de l'entité électronique.

L'entité électronique peut être en pratique une carte à microcircuit, particulièrement adaptée à ce type d'opérations sécurisées.
25

Dans un mode de réalisation, ladite détermination d'une donnée de sortie comprend:

- la détermination d'un mot d'une table de conversion en fonction d'une partie de ladite donnée d'entrée et d'une partie correspondante de ladite
30 valeur de masquage, ledit mot étant composé d'une pluralité de données de sortie de ladite table de conversion et comprenant ladite donnée de sortie;

- l'accès à ladite donnée de sortie dans ledit mot à partir de l'autre partie de la donnée d'entrée et de l'autre partie de la valeur de masquage.

En particulier, ledit accès à la donnée de sortie dans ledit mot comprend:

- 5 i) une étape de séparation dudit mot en deux moitiés respectivement des bits les plus significatifs et des bits les moins significatifs,
- ii) une étape de sélection d'une desdites moitiés dudit mot en fonction des valeurs des bits d'un même index dans respectivement l'autre partie de la donnée d'entrée et l'autre partie de la valeur de masquage.

10 L'invention propose également un dispositif électronique de traitement cryptographique de données représentées sous forme numérique, le dispositif étant apte à effectuer une transformation d'une donnée d'entrée, masquée par au moins un premier masque d'entrée, en une donnée de sortie, ladite transformation mettant en œuvre une table de conversion représentant

15 une fonction non linéaire, caractérisé par

- des moyens de comparaison entre une valeur de masquage et ledit premier masque d'entrée;
- des moyens de détermination, à l'aide de ladite table de conversion, d'une donnée de sortie correspondant à la donnée d'entrée
- 20 masquée à laquelle est appliquée ladite valeur de masquage;
- des moyens agencés pour stocker, dans un registre dépendant du résultat de ladite comparaison, de la donnée de sortie déterminée.

Ce dispositif peut également inclure des caractéristiques correspondant aux modes de réalisation envisagés ci-dessus pour le procédé.

25 L'invention propose enfin un produit programme d'ordinateur comprenant une suite d'instructions aptes, lorsqu'elles sont exécutées par un microprocesseur, à mettre en œuvre le procédé évoqué ci-dessus.

Les avantages, buts et caractéristiques particulières de ce dispositif électronique et de ce produit programme d'ordinateur étant similaires à ceux du

30 procédé objet de la présente invention, telle que succinctement exposée ci-dessus, ils ne sont pas rappelés ici.

D'autres avantages, buts et caractéristiques particulières de la présente invention ressortiront de la description qui va suivre, faite, dans un but explicatif et nullement limitatif en regard des dessins annexés, dans lesquels :

- la **figure 1** représente schématiquement les éléments principaux d'une forme de réalisation possible pour une carte à microcircuit ;
 - la **figure 2** représente l'allure physique générale de la carte à microcircuit de la figure 1 ;
 - la **figure 3** représente, sous forme de logigramme, les étapes essentielles d'un chiffrement selon l'algorithme AES avec masquage ;
 - la **figure 4** représente, sous forme de logigramme, un premier mode de réalisation de l'invention mis en œuvre dans le procédé de la **figure 3**;
 - la **figure 5** représente, sous forme de logigramme, un deuxième mode de réalisation de l'invention mis en œuvre dans le procédé de la **figure 3**;
 - la **figure 6** représente, sous forme de logigramme, un troisième mode de réalisation de l'invention mis en œuvre dans le procédé de la **figure 3**;
 - la **figure 7** illustre, sous forme de logigramme, un mode de réalisation de la fonction *compare* utilisée dans les différents modes de réalisation des **figures 4 à 6**;
 - les **figures 8a et 8b** illustrent un exemple de mise en œuvre des tables-S utilisées dans les modes de réalisations des **figures 4 à 7**;
 - la **figure 9** représente, sous forme de logigramme, un premier mode de réalisation pour l'accès à un sous-mot d'un mot binaire, par exemple stocké dans la table-S de la **figure 8b**;
 - la **figure 10** représente, sous forme de logigramme, un deuxième mode de réalisation pour l'accès à un sous-mot d'un mot binaire, par exemple stocké dans la table-S de la **figure 8b**; et
 - la **figure 11** représente, sous forme de logigramme, un troisième mode de réalisation pour l'accès à un sous-mot d'un mot binaire, par exemple stocké dans la table-S de la **figure 8b**.
- On notera que les modes de réalisation relatifs aux **figures 1 à 7** et les réalisations des **figures 8 à 11** constituent des avancées indépendantes qui peuvent faire l'objet, chacune, d'une protection spécifique.

Un exemple d'entité électronique est une carte à microcircuit 10 dont les principaux éléments électroniques sont représentés à la **figure 1** et qui comporte un microprocesseur 2 relié d'une part à une mémoire vive (ou RAM de l'anglais *Random Access Memory*) 4 et d'autre part à une mémoire à semi-conducteur non-volatile réinscriptible 6, par exemple une mémoire morte effaçable et programmable électriquement (ou EEPROM de l'anglais *Electrically Erasable Programmable Read Only Memory*). En variante, la mémoire non-volatile réinscriptible à semi-conducteur 6 pourrait être une mémoire flash.

Les mémoires 4, 6 sont reliées au microprocesseur 2 par un bus chacune sur la **figure 1** ; en variante, il pourrait s'agir d'un bus commun.

La carte à microcircuit 10 comporte également une interface 8 de communication avec un terminal utilisateur réalisée ici sous forme de contacts dont un assure par exemple une liaison bidirectionnelle avec le microprocesseur 2. L'interface 8 permet ainsi l'établissement d'une communication bidirectionnelle entre le microprocesseur 2 et le terminal utilisateur dans lequel la carte à microcircuit 10 sera insérée.

Ainsi, lors de l'insertion de la carte à microcircuit 10 dans un terminal utilisateur, le microprocesseur 2 va mettre en œuvre un procédé de fonctionnement de la carte à microcircuit 10, selon un jeu d'instructions, stockées par exemple dans une mémoire morte (ou ROM de l'anglais *Read-Only Memory*) – non représentée – ou dans la mémoire réinscriptible 6, qui définit un programme d'ordinateur. Ce procédé inclut en général l'échange de données avec le terminal utilisateur *via* l'interface 8 et le traitement de données au sein de la carte à microcircuit 10, et précisément au sein du microprocesseur 2 avec utilisation éventuelle de données stockées dans la mémoire réinscriptible 6 et de données stockées temporairement dans la mémoire vive 4.

Des exemples de procédés qui mettent en œuvre l'invention sont donnés dans la suite.

La **figure 2** représente l'allure physique générale de la carte à microcircuit 10 réalisée avec la forme générale d'un parallélépipède rectangle de très faible épaisseur.

L'interface de communication 8 pourvue des contacts déjà mentionnés apparaît clairement sur la face de la carte à microcircuit 10 visible sur la **figure 2**, sous forme d'un rectangle inscrit dans la face supérieure de la carte à microcircuit 10.

5 On décrit maintenant des exemples de réalisation de l'invention en référence à un algorithme cryptographique de type AES ("*Advanced Encryption Standard*" selon la terminologie anglo-saxonne) représenté synthétiquement sur la **figure 3**.

10 Il est néanmoins entendu que l'invention peut par exemple être appliquée dans le cas d'autres algorithmes impliquant une fonction non-linéaire, tel l'algorithme DES, avec par exemple l'utilisation de blocs de chiffrement (également connus sous la terminologie "*block cipher*"). Le masque ou les masques appliqués à la donnée en entrée de la fonction non-linéaire ne sont d'ailleurs pas nécessairement le ou les masques X appliqués au début de
15 l'algorithme, mais s'en déduisent en général facilement selon les mécanismes de l'algorithme, par exemple tel que mentionné dans la demande WO 2007/116140 susvisée.

La **figure 3** représente les étapes essentielles du procédé de chiffrement AES d'un mot M au sein de l'entité électronique.

20 Le mot M est en général une partie d'un message à chiffrer ayant par exemple une longueur de 128 bits. D'autres longueurs sont naturellement envisageable, telles que les longueurs de 192 bits et 256 bits fréquemment utilisées.

L'exemple décrit ici utilise comme entité électronique la carte à
25 microcircuit décrite précédemment en référence aux **figures 1** et **2**, mais d'autres types d'entité électronique peuvent naturellement être utilisés, tels que par exemple un ordinateur personnel.

Pour ce faire, l'entité électronique mémorise, par exemple au sein de la mémoire non volatile 6, une clé cryptographique K à partir de laquelle sont
30 dérivées des sous-clés $K_0, \dots, K_n$ au moyen d'une procédure d'expansion de clé.

L'obtention des sous-clés $K_0, \dots, K_n$ peut être réalisée selon des techniques connues et qui ne seront donc pas décrites en détail ici. On pourra se référer par exemple à la demande de brevet FR 2 838 262.

On notera toutefois que l'invention, décrite dans la suite pour le calcul
5 au niveau des boîtes-S (ou "S-box") impliquées à chaque itération (ou ROUND) de l'algorithme AES, pourrait être mise en œuvre lors de l'application de la fonction non-linéaire utilisée dans l'algorithme de dérivation des sous-clés $K_0, \dots, K_n$.

Le procédé de chiffrement commence à l'étape E100 par la réception,
10 par exemple à travers l'interface 8 de la carte à microcircuit, du mot (en général une partie de message) M à chiffrer.

On procède alors au sein de l'entité électronique au tirage d'un nombre aléatoire X utilisé comme masque du mot M au cours d'une étape E102. Bien que l'on utilise les termes "*nombre aléatoire*", il s'agit par exemple en pratique d'un
15 nombre pseudo-aléatoire déterminé à l'intérieur du microprocesseur 2. De manière générale, le nombre X et toute donnée référencée ci-après comme aléatoire doit être un nombre non-prévisible de l'extérieur de l'entité électronique.

On procède alors au cours d'une étape E104 au masquage du mot M afin d'obtenir un mot masqué M' par combinaison du mot M et du nombre aléatoire X au moyen d'une opération de OU EXCLUSIF (fréquemment
20 dénommée "XOR") : $M' = M \oplus X$.

Lorsqu'un simple masque est utilisé pour masquer une donnée (masquage d'ordre 1), la fuite d'information sur la donnée masquée peut-être
25 ciblée conjointement avec la fuite d'information sur le masque afin de retrouver de l'information sur la donnée en claire. Ce type d'attaques couramment appelées attaques d'ordre 2 peut être évité par l'utilisation d'un second masque (masquage d'ordre 2).

Ainsi en variante, ce peut être deux masques X_1 et X_2 qui sont générés pour doublement masquer le message M : $M' = (M \oplus X_1) \oplus X_2$. Cette configuration
30 fournit une protection du message M au deuxième ordre de fuite. On décrira notamment par la suite un exemple de réalisation de l'invention impliquant ce double masquage.

On affectera dans la suite le symbole prime aux valeurs masquées (ici par exemple M') tandis que les grandeurs qui ne portent pas le symbole prime représentent les grandeurs correspondantes sans masquage, c'est-à-dire telles qu'elles auraient été obtenues au cours de l'algorithme AES réalisé sans masquage ; ces grandeurs sans masquage sont introduites ici à des fins explicatives, mais ne seront pas manipulées par le procédé décrit ici qui utilise leur version masquée, sauf naturellement pour les grandeurs M et M_n utilisées respectivement en entrée et en sortie.

Il va à présent être procédé aux différentes étapes de l'algorithme AES en utilisant la donnée masquée M' selon les étapes décrites ci-dessous qui correspondent aux étapes classiques de l'algorithme AES adaptées pour tenir compte du masquage.

On procède ainsi tout d'abord à la transformation initiale au moyen de la sous-clé K_0 au cours d'une étape E106 en appliquant la clé K_0 à la donnée au moyen d'un opérateur OU EXCLUSIF: $M'_0 = M' \oplus K_0$.

Si on dénomme M_0 le résultat de la transformation initiale sans masquage ($M_0 = M \oplus K_0$), on peut remarquer que le résultat M'_0 de l'étape E106 peut s'écrire $M'_0 = M_0 \oplus X$. On remarque ainsi que le résultat de l'étape E106 correspond au résultat de la transformation initiale sans masquage, masqué par la valeur du masque X .

On procède après la transformation initiale à une étape E108 d'initialisation à 1 d'un indice i repérant dans la suite l'itération (ou ROUND) concernée.

A l'étape E110, on fait subir au mot masqué M'_{i-1} obtenu à l'étape précédente (étape E106 après la transformation initiale ou itération précédente) une itération (ROUND) des étapes E110 et E112, afin d'obtenir un nouveau mot masqué M'_i .

Chaque itération est conçue pour que le résultat obtenu après l'itération M'_i soit égal au résultat M_i après l'itération i dans un algorithme sans masquage, masqué avec un masque X identique à celui introduit à l'étape E104 : après chaque itération, $M'_i = M_i \oplus X$.

En alternative, on peut prévoir que le masquage est modifié à chaque itération, par exemple en démasquant M'_i et en le re-masquant avec un nouveau masque Y. A titre d'exemple, cette opération peut être effectuée à l'étape E210 mentionnée dans la demande WO 2007/116140 susvisée.

5 Une fois l'itération ou ROUND i réalisée, on incrémente la valeur de l'indice i au cours d'une étape E114, puis on teste au cours d'une étape E116 l'égalité $i = n$, où n est le nombre d'itérations plus un utilisées dans l'algorithme concerné (en général 10 itérations pour un mot de 128 bits).

10 Si la dernière itération n'a pas été atteinte (c'est-à-dire que l'égalité $i = n$ n'est pas vérifiée), on retourne à l'étape E110 pour la mise en œuvre de l'itération suivante. A cette occasion, et comme indiqué ci-dessus, on peut générer un nouveau masque et l'utiliser pour masquer le résultat M'_i .

Si en revanche la dernière itération est atteinte (c'est-à-dire lorsque l'on vérifie $i = n$), on procède à la transformation finale lors d'une étape E118, au cours
15 de laquelle on obtient donc un mot M'_n à partir du mot M'_{n-1} précédemment obtenu, et ce avec utilisation de la sous-clé K_n (étape E118).

Le résultat de la transformation finale M'_n correspond ainsi au mot chiffré obtenu grâce à l'algorithme AES à partir du mot initial M , masqué avec le masque précédemment défini X (ou le dernier masque intermédiaire Y généré si
20 l'on souhaite modifier le masquage au cours des itérations).

On procède alors à l'étape E120 au démasquage du mot obtenu à l'étape E118 afin d'obtenir le mot chiffré M_n : $M_n = M'_n \oplus X$ (ou avec le masque Y).

Le mot chiffré M_n peut alors être émis à l'extérieur de l'entité électronique au moyen de l'interface 8 au cours d'une étape E122 qui termine le
25 procédé de chiffage par l'entité électronique du mot M .

La présente invention porte plus précisément sur l'utilisation des SBox lors de l'itération des étapes E110 et E112.

Dans le domaine de la cryptographie, la transformation $M'_{i-1} \rightarrow M'_i$ peut être modélisée par la composition de trois opérations: une fonction additive
30 de la clé dérivée K_i , une fonction non linéaire et une fonction linéaire.

L'étape E110 illustre l'application de la fonction non linéaire par l'utilisation d'une boîte-S et décrite plus en détail en référence aux modes de réalisation ci-après.

5 L'étape E112 correspond alors à l'application des deux autres fonctions dont un exemple est proposé dans la demande WO 2007/116140 susvisée en lien avec sa figure 4, notamment des étapes de décalage de bits de sous-blocs (également appelée étape de *Shift Row*), de multiplication par une matrice (étape appelée *Mix Column*) et d'addition de la clé K_i (étape *Add Round Key*).

10

Pour la réalisation de l'étape E110, la table-S est stockée dans la mémoire non volatile 6, table également dénommée table de conversion (*look-up table* ou LUT). La table reçoit des données d'entrée de dimension m et fournit des données de sortie de dimension n , notamment m peut être égale à 15 n . L'invention s'applique également dans le cas où ces deux dimensions sont différentes.

On dispose également de deux registres R_0 et R_1 dont l'utilisation est décrite ci-après. Ces registres sont notamment des registres binaires aptes à stocker des données numériques de sortie de la table-S (c'est-à-dire qu'ils ont 20 une dimension appropriée au moins égale à n permettant ce stockage). En particulier, ces registres sont prévus dans le microcontrôleur de la carte 10 voire dans le microprocesseur 2.

On a représenté un premier mode de réalisation de l'étape E110 à la **figure 4**.

25 A l'étape E200, on dispose de la donnée masquée M'_{i-1} et du masque X , tous deux représentés sous forme binaire de dimension m .

A l'étape E202, on initialise une variable binaire de masquage A à 0, qui sert à définir des itérations successives d'accès à la table S . Cette valeur A prend des valeurs entre 0 et 2^m-1 et est donc de même dimension que M'_{i-1} et 30 X .

A l'étape E204, on procède à la comparaison de la variable binaire de masquage A de l'itération en cours, en l'espèce la valeur 0 pour la première

itération, avec le masque X d'entrée. Pour ce faire, on utilise une fonction booléenne de comparaison définie par:

$$compare(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

Ainsi, quelque soit l'itération considérée, si $A=X$ alors la comparaison
5 fournit la valeur $cmp=0$, sinon la valeur 1 est retournée.

En l'espèce si X est différent du masque 0, alors la fonction de comparaison retourne 1.

A l'étape E206, on utilise la table S pour calculer la donnée de sortie masquée B' de SBox associée à $M'_{i-1} \oplus A$, notamment masquée par un masque
10 binaire de sortie Z de dimension n: $B' = S(M'_{i-1} \oplus A) \oplus Z$.

On stocke cette valeur binaire B' dans le registre d'indice *cmp* calculé précédemment à l'étape E204.

On remarquera que la valeur B' est stockée dans le registre R_0 uniquement si $A=X$, et dans le registre R_1 si $A \neq X$.

15 On incrémente la valeur A de une unité binaire à l'étape E208.

A l'étape E210, on compare la valeur de masquage A à 2^m , représentant le nombre d'entrées de la table S.

En cas de comparaison négative, on procède à une nouvelle itération des étapes E204 à E210 qui permettent d'accéder à une nouvelle entrée de la
20 table S.

En cas de comparaison positive à l'étape E210, on a parcouru l'ensemble des entrées de la table S et on met fin au processus de calcul de la SBox en retournant (étape E212) la valeur stockée dans le registre R_0 , éventuellement accompagnée du masque de sortie Z.

25 On remarque que le registre R_0 contient la valeur $S(M'_{i-1} \oplus A) \oplus Z$ lorsque $A=X$, à savoir $S(M'_{i-1} \oplus X) \oplus Z = S(M_{i-1} \oplus X \oplus X) \oplus Z = S(M_{i-1}) \oplus Z$. On a ainsi, sans manipulé directement la valeur M_{i-1} non masquée, obtenu sa transformée (masquée au premier ordre) par la fonction non linéaire de la SBox.

30 On note, par ailleurs, que cette réalisation met en œuvre des opérations factices ou inutiles (*dummy* selon la terminologie anglo-saxonne) au

sens où elles ne sont pas prises en compte dans le calcul de la valeur finale contenue dans R_0 , c'est le cas de tous les calculs stockés dans R_1 .

Afin d'éviter la présence de telles opérations factices éventuellement détectables par un attaquant mal intentionné pour déterminer la seule opération non factice, on peut proposer de modifier la méthode ci-dessus par les éléments suivants, illustrés par la **figure 5**.

On part des mêmes éléments (étape E300). La table S représente une fonction F non linéaire dite équilibrée ("*balanced*") au sens où elle vérifie:

$$\text{cardinal}[F^{-1}(y)] = 2^{m-n} \text{ pour tout élément } y \text{ binaire de longueur } n.$$

De telles fonctions équilibrées sont toujours utilisées dans les applications cryptographiques requerrant un niveau de sécurité important.

Lors de l'initialisation de l'étape E302, on initialise, outre A à 0, les registres R_0 et R_1 à, chacun, la valeur Z du masque de sortie. En présence de plusieurs masques, il y aura lieu de stocker la combinaison des masques de sorties, par exemple $Z_1 \oplus Z_2$.

A l'étape E304, on compare A et X de façon similaire à l'étape E204 ci-dessus.

A l'étape E306, on augmente le registre R_{cmp} de la valeur SBox de sortie calculée pour le masque A: $R_{\text{cmp}} \leftarrow R_{\text{cmp}} \oplus S(M'_{i-1} \oplus A)$.

On procède ainsi pour chacune des valeurs de masquage A de 0 à $2^m - 1$ (étapes E308, E310 similaires aux étapes E208 et E210).

Au sortir des itérations, on compare, à l'étape E312, les deux registres: $\text{cmp} \leftarrow \text{compare}(R_0, R_1)$. Cette étape retourne 0 si les deux registres sont égaux et 1 s'ils sont différents.

A l'étape E314, on retourne comme donnée de sortie masquée de la SBox, la valeur $R_0 \oplus (\text{cmp} \times R_1)$.

Au sortir de l'algorithme, le registre R_1 comprend la valeur $Z \oplus \bigoplus_{\substack{A \in F_2^m \\ A \neq X}} S(M'_{i-1} \oplus A)$. Du fait des propriétés de la fonction F de la SBox (fonction *balanced*), cette valeur est égale, dans une exécution normale, à $S(M'_{i-1} \oplus X) \oplus Z$, c'est-à-dire identique à R_0 .

Par conséquent, la comparaison de l'étape E312 doit retourner 0 dans une exécution normale et l'algorithme retourne alors $R_0 = S(M'_{i-1} \oplus X) \oplus Z$.

Le résultat E314 est alors dépendant de toutes les étapes exécutées, apportant un gain de sécurité.

5 En référence à la **figure 6**, on a représenté un autre mode de réalisation de l'invention reposant sur l'affectation (pseudo)aléatoire des registres et sur une protection contre les fuites de deuxième ordre. Il est entendu que l'affectation des registres peut également s'appliquer au mode de réalisation correspondant à la **figure 4**. De façon symétrique, la mise en œuvre
10 décrite en référence à la **figure 5** peut également être réalisée par l'utilisation d'un double masquage (voire plus) et l'affectation des registres.

A l'étape E400, on dispose de la donnée doublement masquée M'_{i-1} et des masques X_1 et X_2 ($M'_{i-1} = (M_{i-1} \oplus X_1) \oplus X_2$), tous représentés sous forme binaire de dimension m .

15 A l'étape E402, on génère (pseudo)-aléatoirement un booléen b , qui prend donc soit la valeur 0 soit la valeur 1. On utilise ce même booléen pour toute l'exécution du calcul sur la SBox.

A l'étape E404, on initialise, de façon similaire à celle indiquée ci-dessus, la variable binaire de masquage A à 0, qui sert à définir des itérations
20 successives d'accès à la table S .

A l'étape E406, on compare la valeur combinée $X_1 \oplus A$ des masques X_1 et A , avec le masque d'entrée X_2 . Pour ce faire, on utilise la fonction $compare_b$ qui dérive de la fonction $compare$, identifiée précédemment, comme suit:

$$25 \quad compare_b(x, y) = \begin{cases} b & \text{if } x = y \\ \bar{b} & \text{if } x \neq y \end{cases}$$

Une réalisation de cette fonction (également applicable à la fonction $compare$ simple) est décrite ci-après en référence à la **figure 7**.

On note ici que l'on combine X_1 avec A afin de ne pas manipuler directement $X_1 \oplus X_2$, ce qui correspondrait à une protection d'ordre un ($X_1 \oplus X_2$
30 pouvant être vu comme un unique masque d'entrée).

A l'étape E408, on utilise, de façon similaire à l'étape E206 ci-dessus, la table S pour calculer la donnée de sortie masquée B' de SBox associée à $M'_{i-1} \oplus A$, notamment masquée par deux masques binaires de sortie Z_1 et Z_2 de dimension n : $B' = S(M'_{i-1} \oplus A) \oplus Z_1 \oplus Z_2$.

5 On stocke cette valeur binaire B' dans le registre d'indice *cmp* calculé précédemment à l'étape E406.

On remarquera que la valeur B' est stockée dans le registre R_b uniquement si $A = X_1 \oplus X_2$, et dans l'autre registre si $A \neq X_1 \oplus X_2$. On remarquera également que, du fait de la détermination aléatoire de la valeur b au début d'un calcul par SBox, ce n'est pas toujours dans le même registre que l'on stocke la donnée de sortie correspondant à la combinaison des masques d'entrée ($A = X_1 \oplus X_2$).

On incrémente la valeur A de une unité binaire à l'étape E410.

15 A l'étape E412, on compare la valeur de masquage A à 2^m , représentant le nombre d'entrées de la table S.

En cas de comparaison négative, on procède à une nouvelle itération des étapes E406 à E412 qui permettent d'accéder à une nouvelle entrée de la table S.

20 En cas de comparaison positive à l'étape E412, on a parcouru l'ensemble des entrées de la table S et on met fin au processus de calcul de la SBox en retournant la valeur stockée dans le registre R_b , éventuellement accompagnée des masques de sortie Z_1 et Z_2 .

On remarque que le registre R_b contient la valeur $S(M'_{i-1} \oplus A) \oplus Z_1 \oplus Z_2$ lorsque $A = X_1 \oplus X_2$, à savoir

25 $S(M'_{i-1} \oplus X_1 \oplus X_2) \oplus Z_1 \oplus Z_2 = S(M_{i-1} \oplus X_1 \oplus X_2 \oplus X_1 \oplus X_2) \oplus Z_1 \oplus Z_2 = S(M_{i-1}) \oplus Z_1 \oplus Z_2$.

On a ainsi, sans manipulé directement la valeur M_{i-1} non masquée obtenue sa transformée (masquée au premier ordre) par la fonction non linéaire de la SBox.

30 On note que les comparaisons impliquant la valeur de masquage A et réalisées ci-dessus, peuvent être mises en œuvre pas une comparaison bit à bit des deux paramètres des fonctions *compare*.

Néanmoins, la **figure 7** propose une réalisation alternative de la fonction *compare_b*, également applicable à la fonction *compare*, utilisant une table T de valeurs booléennes de 2^m entrées (même nombre d'entrées que la table-S et que les valeurs possibles de A). La fonction de comparaison est ainsi
 5 mise en œuvre par un simple accès en lecture à la table T.

Pour ce faire, lors d'une étape d'initialisation, par exemple lors de l'étape E402, on procède à la génération aléatoire d'un index r3: $r3 \leftarrow \text{rand}(m)$ (étape 500).

A l'étape 502, on initialise la table T avec la valeur $\bar{b}$ (où $\bar{b}$ est le
 10 complémentaire de b) à chacune des entrées à partir du booléen généré à l'étape E402: $T \leftarrow \{\bar{b}; \bar{b}; \bar{b}; \dots; \bar{b}\}$.

A l'étape 504, on affecte la valeur du booléen b à l'entrée correspondant à l'index généré r3: $T[r3] \leftarrow b$.

Ainsi, à la fin du processus d'initialisation, on a une table T
 15 configurée comme suit:

$$T[x] = \begin{cases} b & \text{if } x = r3 \\ \bar{b} & \text{sinon} \end{cases}$$

Lors de l'utilisation de la fonction de comparaison (étapes E204, E304, E406), on reçoit deux valeurs α et β à comparer (étape E506) correspondant généralement à la donnée de masquage (éventuellement
 20 combinée à un masque) et à un masque d'entrée.

La comparaison consiste alors à l'étape E508 à retourner l'entrée du tableau d'indice $(\alpha \oplus r3) \oplus \beta$. En effet, si $\alpha = \beta$, alors le processus retourne $T[r3]=b$.

Ainsi, l'invention telle que décrit ci-dessus permet de sécuriser le
 25 calcul d'une sortie de SBox masquée à partir d'une entrée masquée pour les ordres de masquages traditionnellement utilisés, à savoir aussi bien pour un masquage de premier ordre qu'un masquage de deuxième ordre.

Quel que soit le mode de réalisation envisagé ci-dessus,
 30 l'architecture matérielle des équipements électroniques, tels que le microcircuit 10, impose parfois l'utilisation de mots machine sur un nombre de bits

déterminés, par exemple 8 bits, 16 bits ou 32 bits. Ainsi, lorsque les données de sortie sont des mots d'un nombre inférieur de bits, par exemple 4 bits, on mémorise plusieurs mots de sortie $S(M)$ dans un seul mot machine.

Les **figures 8a** et **8b** illustrent cette réalisation pratique. Sur la **figure 8a**, on observe la table S utilisée ci-dessus présentant 2^m mots de sortie de longueur n . Sur la **figure 8b**, la réalisation pratique mémorise les mots de sortie de longueur n , ici 2 bits, sur des mots machines de longueur $2^w n$, ici 16 bits (w est alors égal à 3). Le stockage de l'ensemble des mots de sortie $S(M)$ ne requiert alors que l'utilisation de 2^{m-w} mots machines, stockant chacun respectivement 2^w mots de sortie $S(M)$ (également appelés par la suite "sous-mots"). On obtient ainsi un stockage en mémoire efficace. La table ainsi formée comprend 2^{m-w} mots de longueur $2^w n$.

Les mots de sortie sont mémorisés de sorte qu'on accède au mot $S(M)$ en sélectionnant un mot machine $S_{\text{machine}}(M_H)$ à partir des $m-w$ bits les plus significatifs de M (notés pour la suite M_H) et en récupérant le sous-mot du mot sélectionné à partir des w bits les moins significatifs de M (ci-après notés M_L). Notamment $S_{\text{machine}}(M_H) = \{S(M_H, 0), S(M_H, 1), \dots, S(M_H, 2^w - 1)\}$.

Une autre convention qui décompose le mot M peut également être envisagée, par exemple sélectionner le mot machine à partir des bits centraux de M et sélectionner le sous-mot à partir de bits extrêmes de M (par exemple les deux bits de poids fort et le bit de poids faible).

Au vu de ce qui précède, on peut calculer le mot machine $S_{\text{machine}}(M_H)$ de manière sécurisée en appliquant les algorithmes, évoqués ci-dessus en lien avec les **figures 4 à 7**, aux parties de poids fort de M' et de son masque X .

On récupère ainsi $S_{\text{machine}}(M_H)$ éventuellement masqué, en fonction de $M'_{i-1,H}$, A_H ou X_H (éventuellement $X_{1,H}$ et $X_{2,H}$ lors du double masquage) (l'indice H représentant les $m-w$ bits les plus significatifs).

Sans masquage de sortie, on récupère $S_{\text{machine}}(M_H)$.

Avec un masquage de sortie simple à l'aide du masque Z , on récupère $S_{\text{machine}}(M_H) \oplus Z$ (Z de longueur $m-w$).

Avec un masquage de sortie double à l'aide des masques Z_1 et Z_2 , on récupère $S_{\text{machine}}(M_H) \oplus Z_1 \oplus Z_2$ (Z_1 et Z_2 de longueur $m-w$).

Des ordres de masquage supérieurs peuvent également être envisagés sans complexifier les mécanismes décrits par la suite.

5 De l'autre côté, on dispose de $M'_{i-1,L}$, X_L (éventuellement $X_{1,L}$ et $X_{2,L}$ lors du double masquage) qui correspondent aux bits d'identification du mot de sortie à l'intérieur du mot machine $S_{\text{machine}}(M_H)$, éventuellement masqué.

On décrit maintenant, en référence aux **figures 9 à 11**, différents mécanismes permettant d'extraire efficacement, et éventuellement en toute
10 sécurité eu égard aux degrés de masquage, le sous-mot (donc le mot de sortie) d'indice $M_{i-1,L}$ du mot machine $S_{\text{machine}}(M_H)$ éventuellement masqué en sortie, à partir de l'indice masqué $M'_{i-1,L}$ et des masques A_L ou X_L et éventuellement $X_{1,L}$ et $X_{2,L}$.

Pour simplifier la notation, on note désormais U (ou U' si masqué) le
15 mot machine, j' l'indice masqué, r_j (r_{j1} et r_{j2}) les masques de l'indice, r_u (r_{u1} et r_{u2}) les masques de sorties masquant éventuellement le mot machine U' . On cherche désormais à extraire le sous-mot $U(j)$ à partir de j' et des masques de j' (ou extraire $U'(j)$ et les masques de sorties correspondant notés s_u (s_{u1} et s_{u2})), c'est-à-dire sans manipuler j pour des questions de sécurité.

20 Ces différents mécanismes concernent généralement l'accès à un sous-mot d'indice j dans un mot binaire (U , U') formé de 2^w sous-mots $\{U(0), \dots, U(2^w-1)\}$ à partir de l'indice binaire j' masqué par un masque binaire correspondant r_j , comprenant

i) une étape de séparation dudit mot (U , U') en deux moitiés ($H_0(U)$, $H_1(U)$, U_H ,
25 U_L) respectivement des bits les plus significatifs et des bits les moins significatifs,

ii) au moins une étape de sélection d'une moitié dudit mot (U , U') en fonction des valeurs des bits d'un même index dans respectivement l'indice j' et le masque r_j .

30 En effet, on remarque que selon la valeur du bit de poids fort de j , le sous-mot auquel on veut accéder est soit dans la partie gauche (c'est-à-dire de poids fort), soit dans la partie droite (c'est-à-dire de poids faible) de U . Ainsi, en

combinant l'utilisation des bits respectifs de l'indice masqué et de son masque (c'est-à-dire des bits contribuant à obtenir le bit correspondant de l'indice j non masqué), on sélectionne efficacement la partie du mot d'intérêt, sans manipuler l'indice j non masqué.

5 En particulier, on choisira des sous-mots de même longueur.

On illustre une première réalisation de ces mécanismes à l'aide du mot $U=1010011101101101$ composé de 8 sous-mots d'indice allant de 000 (sous-mot de poids fort – à gauche) à 111 (sous-mot de poids faible – à droite), en s'appuyant sur la **figure 9**. Plus précisément, on souhaite accéder au sous-mot dont l'indice masqué j' vaut 101 et son masque r_j 111 (on rappelle $j'=j\oplus r_j$). On remarque que $j=101\oplus 111=010$, donc on souhaite accéder au sous-mot 01 composé des 5^è et 6^è bits de U en partant de la gauche.

Cette réalisation met en oeuvre les étapes algorithmiques suivantes:

for $k=0$ to $w-1$

15 $(R_0, R_1) \leftarrow U$
 $\text{swap}_{j'[k]}(R_0, R_1)$
 $\text{swap}_{r_j[k]}(R_0, R_1)$
 $U \leftarrow R_0$
 return R_0

20 où R_0 et R_1 sont deux registres de longueur au moins égale à la demi-longueur de U , ici au moins 8 bits, et $\text{swap}_b(R_0, R_1)$ est une fonction qui intervertit le contenu des deux registres lorsque $b=0$:

$$\text{swap}_b(R_0, R_1) = \begin{cases} \text{NOP} & \text{si } b = 0 \\ \text{swap}(R_0, R_1) & \text{si } b = 1 \end{cases}$$

Une telle fonction swap_b peut être mise en oeuvre par les étapes

25 suivantes: $\text{tmp} \leftarrow R_1$
 $R_b \leftarrow R_0$
 $R_{\bar{b}} \leftarrow \text{tmp}$

En alternative, swap_b peut être mise en oeuvre à l'aide de la fonction $\text{Rotate}(R, x)$ qui effectue une rotation de R sur x bits à droite ou à gauche, par
 30 $\text{swap}_b(R_0, R_1) = \text{Rotate}((R_0, R_1), l+b.l)$ où l est la longueur des registres. Ici, on utilisera un seul registre R composé des deux parties égales R_0 et R_1 , afin que

le décalage de bits décale potentiellement les bits d'une partie du registre R à l'autre.

En alternative à l'utilisation de deux registres, il est possible d'utiliser un registre double (longueur $2l$, ici au moins 16 bits) constitué de deux parties.

5 A l'étape E600, on initialise un compteur k à 0.

A l'étape E602, on affecte aux deux registres R_0 et R_1 , respectivement U_H (10100111) et U_L (01101101).

A l'étape E604, on applique la fonction $swap_b$ en fonction du bit le plus significatif de l'indice $j'=\{j'[0], j'[1], \dots, j'[w-1]\}$:

10 $j'[k=0]=1$.

Ici, on intervertit alors les deux registres: $R_0 = 01101101$ et $R_1 = 10100111$.

A l'étape E606, on applique de nouveau la fonction $swap_b$ en fonction du bit de même index k , donc le plus significatif, du masque r_j : $r_j[k=0]=1$.

15 Ici, on intervertit alors les deux registres: $R_0 = 10100111$ et $R_1 = 01101101$.

A l'étape E608, on remplace la valeur de U par le contenu du registre R_0 .

A l'étape E610, on incrémente la valeur d'itération k : $k=1$.

20 A l'étape E612, on compare k avec w (ici $w=3$). Comme $k<3$, on retourne à l'étape E602 en répartissant la nouvelle valeur de U dans les deux registres: $R_0 = 1010$ et $R_1 = 0111$.

Comme $j'[k=1]=0$ et $r_j[k=1]=1$, on effectue une seule permutation des deux registres lors de l'étape E606: $R_0 = 0111$ et $R_1 = 1010$.

25 On conserve ainsi à l'étape E608, $U=0111$.

A l'étape E612, $k=2$, donc on retourne à l'étape E602.

L'itération $k=2$ mène à $R_0 = 01$ et $R_1 = 11$ car deux permutations ont été effectuées aux étapes E604 et E606.

30 A l'étape E612, $k=3=w$, donc on passe à l'étape E614 par laquelle on retourne $U=01$. On obtient bien le sous-mot attendu.

On remarque qu'à chaque itération, la taille de U diminue de moitié et on converge ainsi par dichotomie vers le sous-mot souhaité.

La sécurité de ce mécanisme est assurée par l'application des fonctions de permutation (inversion des deux registres) à toutes les itérations (avec des paramètres différents) même si aucune permutation n'en découle.

5 D'une façon générale, l'opération de permutation (*swap*) telle qu'envisagée ci-dessus confère, lorsqu'elle est utilisée de façon conditionnelle, un degré de sécurité élevé lorsqu'il s'agit d'identifier (par exemple pour sélection ou isolation) un élément parmi deux. A ce titre, il pourra être envisagé une protection de cette approche indépendante des autres enseignements décrits dans la présente demande.

10 On peut noter qu'en utilisant une indexation de j' (et de ses masques) non plus de gauche à droite mais de droite à gauche, il y a lieu de procéder à une permutation (*swap*) supplémentaire à l'intérieur de chacune des itérations k .

15 Afin de fournir une protection d'ordre 1 de masquage des données, le mode de réalisation décrit ci-dessus peut être étendu à la donnée U' masquée à l'aide d'un masque aléatoire r_u . Dans ce cas, on cherche à déterminer non seulement le sous-mot de U' (comme décrit ci-dessus) qui convient mais également le sous-mot de r_u correspondant.

20 L'accès au sous-mot $r_u(j)$ à partir du mot machine r_u est similaire au mécanisme ci-dessus.

Dans chacune des itérations k , on réalise juste après l'étape E608, des étapes E602', E604', E606' et E608' appliquées au masque r_u , similaires à celles de numéro correspondant E602, E604, E606 et E608.

L'algorithme proposé ci-dessus, devient alors:

25 for $k=0$ to $w-1$
 $(R_0, R_1) \leftarrow U'$
 $\text{swap}_{j'[k]}(R_0, R_1)$
 $\text{swap}_{rj[k]}(R_0, R_1)$
 $U' \leftarrow R_0$
 30 $(R_0, R_1) \leftarrow r_u$
 $\text{swap}_{j'[k]}(R_0, R_1)$
 $\text{swap}_{rj[k]}(R_0, R_1)$

$r_u \leftarrow R_0$
 return (U' , r_u)

On obtient en fin d'algorithme (étape E614) les valeurs U' et r_u correspondant aux sous-mots recherchés, telles que $U' = U \oplus r_u$, où U est la
 5 valeur de sortie (de la boîte S) non masquée.

On décrit maintenant en référence à la **figure 10**, une réalisation plus sécurisée mettant en œuvre un double masquage des valeurs, aussi bien pour l'indice j' (masques r_{j1} et r_{j2}) que pour le mot U' (masques r_{u1} et r_{u2}), par
 10 exemple résultant de l'étape E414 ci-dessus appliquée à la détermination de $S_{\text{machine}}(M_H)$. Ces doubles masquages peuvent bien sûr être appliqués indépendamment.

Cette réalisation diffère, de celle de la **figure 9**, en substance par l'application de la fonction *swap* trois fois en fonction de bits correspondant
 15 dans l'indice j' et dans chacun des deux masques r_{j1} et r_{j2} (on s'aperçoit donc que l'invention peut être étendue sans difficulté à des ordres supérieurs de masquage en effectuant un nombre de permutations égal à $1 + \text{nombre de masques de } j$) et par la détermination des sous-mots dans le mot U' et dans chacun des masques r_{u1} et r_{u2} de U . Pour une homogénéité de sécurité, il
 20 convient de choisir un même degré de masquage de l'indice j et du mot binaire U .

Les étapes E700 à E704 ne diffèrent pas des étapes E600 à E604.

Puisque l'on a deux masques d'indice r_{j1} et r_{j2} , on effectue deux permutations conditionnelles aux étapes E706 et E708, indexées
 25 respectivement sur les deux bits correspondants des deux masques $r_{j1}[k]$ et $r_{j2}[k]$.

L'étape E710 est la même que l'étape E608 ci-dessus.

Dans la même itération, on calcule de façon similaire les deux parties de masques r_{u1} et r_{u2} correspondant, aux étapes E702' à E710' et E702" à
 30 E710".

A l'étape E712, on incrémente k de sorte à former w itérations (comparaison à l'étape E714).

A l'étape E716, on retourne U' , r_u1 et r_u2 représentant les sous-mots souhaités (donc valeurs de sortie de la table S) des mots initiaux. Ces trois valeurs vérifient: $U' = U \oplus r_u1 \oplus r_u2$.

- 5 Afin d'accroître encore plus la sécurité, notamment en évitant d'utiliser les variables intermédiaires U' et $U'(j)$ qui constituent une faiblesse potentielle du fait de la dépendance à l'indice j , on propose le mécanisme suivant en référence à la **figure 11**.

10 Bien que décrit à l'aide d'un double masquage, le mécanisme s'applique à un simple masquage voire en l'absence de masquage du mot U .

 On définit tout d'abord les deux fonctions $H_0(y)$ et $H_1(y)$ qui retournent respectivement la moitié des bits les plus significatifs de y et la moitié des bits les moins significatifs de y . Ces deux fonctions sont aisément réalisables à l'aide de la fonction $swap_b$ évoquée ci-dessus, par exemple
15 comme suit:

$H_b(y) :$ $(R_0, R_1) \leftarrow y$
 $swap_b(R_0, R_1)$
 return R_0

 Le mécanisme d'accès au sous-mot d'indice j peut être mis en œuvre
20 à l'aide des instructions suivantes:
for $k=0$ to $w-1$

$(U', r_u1, r_u2) \leftarrow \text{Select}(2^{w-1-k}n, (U', r_u1, r_u2), (j'[w-k], r_{j1}[w-k], r_{j2}[w-k]))$
return (U', r_u1, r_u2)

 où la fonction Select reçoit en paramètre une dimension *long* ($2^{w-1-k}n$
25 représentant la taille d'une moitié de mot U' , r_u1 et r_u2 que l'on souhaite récupérer en fin d'itération considérée), un premier 3-uplet d'un mot masqué et des deux masques associés, et un deuxième 3-uplet d'un bit masqué (ici le bit d'index $w-k$ de l'indice j) et des deux masques associés, et retourne un 3-uplet
30 (U', r_u1, r_u2) vérifiant $U' \oplus r_u1 \oplus r_u2 = H_{j[w-k]}(U)$.

 On voit ainsi qu'en itérant cette fonction pour chacun des bits composant l'indice j , on isole successivement les moitiés du mot correspondant

aux différents bits de l'indice j pour aboutir au sous-mot de U' d'indice j accompagné des sous-mot des masques correspondants.

On note également que l'application de ce mécanisme à un masquage simple fait intervenir des 2-uplet et non plus des 3-uplet (extensible
5 également à q masques et des q -uplets).

A l'étape E800, on initialise une valeur d'itération k à 0.

Les étapes E802 à E820 illustrent un exemple de réalisation de la fonction Select.

A l'étape E802, on génère deux masques aléatoires de longueur
10 $long=2^{w-1-k}n$.

A l'étape E804, on génère un booléen b aléatoire.

A l'étape E806, on calcule un bit masqué à partir du booléen b et des bits d'index k (en partant du bit de poids le plus fort au bit de poids le plus faible au fur et à mesure des itérations) dans respectivement les deux masques de
15 l'indice j : $b' \leftarrow (r_{j1}[w-k] \oplus b) \oplus r_{j2}[w-k]$

Dans cette étape, on veille à réaliser les opérations XOR dans l'ordre des parenthèses, à savoir d'abord $r_{j1}[w-k] \oplus b$, puis l'addition du résultat avec le booléen $r_{j2}[w-k]$. En effet, en respectant cet ordre, on garantit une protection contre les attaques d'ordre 2 puisque l'on ne manipule pas directement la valeur
20 $r_{j1}[w-k] \oplus r_{j2}[w-k]$ qui pourrait être vue comme un simple masque ne protégeant le bit $j[w-k]$ que par un masquage d'ordre 1.

Pour les étapes suivantes E808 à E818 (qui peuvent être interverties entre elles car indépendantes), on utilise trois paires de registres adressables, ci-après notés A_0 , A_1 (affectés au mot U), B_0 , B_1 (affectés au premier masque de U), C_0 et C_1 (affectés au deuxième masque de U).
25

On affecte successivement:

- au registre A indexé par b' calculé à l'étape E806, la partie $H_{j[w-k]}(U')$ masqué par un (t_1) des masques aléatoires générés à l'étape E802:

$$A_{b'} \leftarrow H_{j[w-k]}(U') \oplus t_1$$

- 30 - à l'autre registre A , l'autre partie de U' masquée par le même masque.

On procède de même pour les deux masques r_u1 et r_u2 , qui utilisent respectivement l'autre masque t_2 et la combinaison des deux masques t_1 et t_2 , et sont stockés par paire respectivement dans B_0 , B_1 , C_0 et C_1 .

A ce stade, on pourrait démontrer que pour l'index ' $w-k$ ' ci-dessus,
5 quelles que soient les valeurs des deux masques $r_{j1}[w-k]$ et $r_{j2}[w-k]$, on a:

$$(A_b, B_b, C_b) = (H_{j[w-k]}(U') \oplus t_1, H_{j[w-k]}(r_u1) \oplus t_2, H_{j[w-k]}(r_u2) \oplus t_1 \oplus t_2)$$

$$(A_{\bar{b}}, B_{\bar{b}}, C_{\bar{b}}) = (H_{\bar{j}[w-k]}(U') \oplus t_1, H_{\bar{j}[w-k]}(r_u1) \oplus t_2, H_{\bar{j}[w-k]}(r_u2) \oplus t_1 \oplus t_2)$$

et les valeurs stockées dans les registres vérifient:

$$A_b \oplus B_b \oplus C_b = H_{j[w-k]}(U).$$

10 Du fait de l'indépendance de ces valeurs à celles de r_{j1} et r_{j2} , on obtient une indépendance par rapport à l'indice j , et donc une sécurité accrue.

Ainsi à l'étape E820, on substitue (A_b, B_b, C_b) aux valeurs précédentes (U', r_u1, r_u2) .

On incrémente ensuite k , à l'étape E822.

15 On compare k et w pour déterminer si on met fin aux itérations, à l'étape E824 (similaire aux étapes E612 et E714 ci-dessus, avec retour à l'étape E802 pour une nouvelle itération).

A la fin des itérations (étape E826), on retourne (U', r_u1, r_u2) qui correspondent respectivement au sous-mot masqué recherché et aux deux
20 sous-mots de masques qui permettent de démasquer le sous-mot U' obtenu: $U' \oplus r_u1 \oplus r_u2 = U$.

L'utilisation des paramètres aléatoires t_1 , t_2 et b n'est pas indispensable. Elle fournit néanmoins un gage de sécurité de l'algorithme car permet, d'une part, de protéger les valeurs manipulées par un double
25 masquage (t_1 et t_2 , que l'on peut réduire à un simple masquage si nécessaire) et d'autre part, d'attribuer le résultat souhaité arbitrairement dans l'un ou l'autre des deux registres indexés sur b . En l'absence de b , on retourne à l'étape E826 A_0 , B_0 et C_0 . En l'absence de t_1 et t_2 , le paramètre *long* n'est pas nécessaire dans la fonction Select indiqué ci-dessus.

30 Dans les mécanismes ci-dessus en lien avec les figures 9 à 11, les valeurs $j'[k]$, $r_j[k]$, $r_{j1}[k]$ et $r_{j2}[k]$ jouent des rôles symétriques de telle sorte qu'il est envisageable d'intervertir leurs positions dans ces mécanismes, par

exemple la valeur b' à l'étape E806 peut utiliser j' au lieu du masque r_{j1} et les fonctions H des étapes E808 et E814 sont alors indexées par le masque r_{j1} .

Grâce à ces mécanismes, on accède de façon sécurisée à sous-mot d'un mot notamment masqué, composé d'une pluralité de sous-mots, à l'aide
5 d'un indice également masqué.

Les exemples qui précèdent ne sont que des modes de réalisation de l'invention qui ne s'y limite pas.

REVENDICATIONS

5 1 - Procédé de traitement cryptographique de données représentées sous forme numérique, le procédé étant mise en œuvre par une entité électronique (10) et comprenant une transformation (E110) d'une donnée d'entrée (M'_{i-1}), masquée par au moins un premier masque d'entrée (X, X_1, X_2), en une donnée de sortie (M'_i), ladite transformation mettant en œuvre une table de conversion (S) représentant une fonction non linéaire (F), le procédé étant caractérisé par les étapes suivantes:

a- comparaison (E204, E304, E406) entre une valeur de masquage (A) et ledit premier masque d'entrée (X, X_1, X_2);

15 b- détermination (E206, E306, E408), à l'aide de ladite table de conversion, d'une donnée de sortie correspondant à la donnée d'entrée masquée à laquelle on applique ladite valeur de masquage;

c- stockage (E206, E306, E408), dans un registre (R_0, R_1) dépendant du résultat de ladite comparaison, de la donnée de sortie déterminée.

20 2 - Procédé selon la revendication 1, dans lequel on itère les étapes a, b et c pour une pluralité de valeurs de masquage (A).

3 - Procédé selon la revendication précédente, dans lequel on itère pour un nombre de valeurs de masquage égale à la taille de la table de conversion de sorte à accéder à chacune des entrées de ladite table de conversion.

25 4 - Procédé selon l'une des revendications précédentes, dans lequel ladite donnée d'entrée est masquée par un premier et un deuxième masques d'entrée (X_1, X_2), ladite étape de comparaison étant alors réalisée entre ledit premier masque d'entrée (X_2) et une combinaison entre ladite valeur de masquage (A) et ledit deuxième masque d'entrée (X_1).

30 5 - Procédé selon la revendication précédente, dans lequel on itère, pour $A=0$ à 2^m-1 , les opérations:

$$cmp \leftarrow compare(X_1 \oplus A, X_2)$$

$$R_{cmp} \leftarrow S(M' \oplus A) \oplus Z_1 \oplus Z_2$$

puis on retourne R_0

où X_1 et X_2 sont deux masques d'entrée, m la dimension d'entrée de la table-S, M' la donnée masquée, R_0 et R_1 deux registres, Z_1 et Z_2 deux masques de sortie et *compare* une fonction retournant 0 en cas d'égalité, 1 sinon.

6 - Procédé selon l'une quelconque des revendications 1 à 4, dans lequel ledit stockage à l'étape c comprend l'addition de la donnée déterminée à la valeur dudit registre.

10 7 - Procédé selon la revendication précédente, comprenant une pluralité d'itérations des étapes a, b et c, et une étape, ultérieure aux itérations, consistant à retourner (E212, E312, E414) une valeur de sortie formée à partir desdits registres.

15 8 - Procédé selon la revendication précédente, dans lequel on itère, pour $A=0$ à 2^m-1 , les opérations suivantes:

$$cmp \leftarrow compare(A, X)$$

$$R_{cmp} \leftarrow R_{cmp} \oplus S(M' \oplus A)$$

puis, postérieurement auxdites itérations on réalise les opérations suivantes:

$$20 \quad cmp \leftarrow compare(R_0, R_1)$$

$$return R_0 \oplus (cmp \times R_1)$$

où X est le masque d'entrée, m la dimension d'entrée de la table-S, M' la donnée masquée, R_0 et R_1 deux registres et *compare* une fonction retournant 0 en cas d'égalité, 1 sinon.

25 9 - Procédé selon la revendication précédente, comprenant, préalablement auxdites itérations, une étape d'initialisation (E302) desdits registres (R_0 , R_1) à une valeur de masquage Z .

30 10 - Procédé selon l'une quelconque des revendications 1 à 9, dans lequel la comparaison a) met en œuvre une fonction (*compare*, *compare_b*) de deux paramètres dont le résultat est un booléen (b) en cas d'égalité entre les deux paramètres et le complémentaire du booléen ($\bar{b}$) sinon, lesdits registres (R_0 , R_1) étant indexés par ledit résultat de la fonction de comparaison.

11 - Procédé selon la revendication précédente, comprenant une étape (E402), préalable auxdites itérations, de génération pseudo-aléatoire dudit booléen (b).

12 - Procédé selon l'une quelconque des revendications précédentes, dans lequel la comparaison (E204, E304, E406) mise en œuvre à l'étape a) pour comparer deux paramètres comprend la lecture (E508) de l'entrée d'une table (T) à un index fonction des deux paramètres (α , β) et d'une valeur prédéterminée (r3).

13 - Procédé selon la revendication précédente, dans lequel la constitution de ladite table (T) comprend:

- la détermination d'un booléen (E402),
- l'initialisation (E502) de chaque entrée de la table (T) avec la valeur du complémentaire dudit booléen,
- la génération pseudo-aléatoire (E500) d'un index r3,
- et l'enregistrement (E504) dudit booléen à l'entrée (T[r3]) de la table (T) correspondant audit index (r3),

ladite comparaison consistant à lire l'entrée $T[(\alpha \oplus r3) \oplus \beta]$ où α et β sont les deux paramètres et T[k] l'entrée d'index k.

14 - Procédé selon l'une quelconque des revendications précédentes, dans lequel lesdits masques (X, X₁, X₂, Z, Z₁, Z₂) sont des masques additifs

15 - Procédé selon la revendication précédente, dans lequel lesdits masques sont des masques booléens.

16 - Procédé selon l'une quelconque des revendications précédentes, comprenant, avec l'étape c, le masquage de ladite donnée de sortie par au moins un premier masque de sortie (Z, Z₁, Z₂).

17 - Procédé selon l'une quelconque des revendications précédentes, dans lequel lesdits registres (R₀, R₁) sont des registres d'un microcontrôleur sécurisé (2).

18 - Procédé selon l'une quelconque des revendications précédentes, dans lequel la table de conversion est mémorisée en mémoire non-volatile.

19 - Procédé selon l'une des revendications 1 à 18, mis en œuvre par une séquence d'instructions mémorisées dans l'entité électronique et exécutées par un microprocesseur de l'entité électronique.

20 - Procédé selon l'une des revendications 1 à 19, dans lequel
5 l'entité électronique est une carte à microcircuit.

21 - Procédé selon l'une des revendications 1 à 20, dans lequel ladite détermination d'une donnée de sortie (E206, E306, E408) comprend:

- la détermination d'un mot ($S_{\text{machine}}(M_H)$) d'une table de conversion en fonction d'une partie de ladite donnée d'entrée (M'_H) et d'une partie
10 correspondante de ladite valeur de masquage (A_H), ledit mot étant composé d'une pluralité de données de sortie de ladite table de conversion et comprenant ladite donnée de sortie;
- l'accès à ladite donnée de sortie dans ledit mot à partir de l'autre partie de la donnée d'entrée (M'_L) et de l'autre partie de la valeur de masquage
15 (A_L).

22 - Procédé selon la revendication précédente, dans lequel ledit accès à la donnée de sortie dans ledit mot comprend:

- i) une étape de séparation dudit mot (U, U') en deux moitiés ($H_0(U), H_1(U), U_H, U_L$) respectivement des bits les plus significatifs et des bits les moins
20 significatifs,
- ii) une étape de sélection d'une desdites moitiés dudit mot (U, U') en fonction des valeurs des bits d'un même index dans respectivement l'autre partie de la donnée d'entrée (M'_L, j') et l'autre partie de la valeur de masquage (A_L, r_j, r_{j1}, r_{j2}).

25 23 - Dispositif électronique de traitement cryptographique de données représentées sous forme numérique, le dispositif étant apte à effectuer une transformation d'une donnée d'entrée (M'_{i-1}), masquée par au moins un premier masque d'entrée (X, X_1, X_2), en une donnée de sortie (M'_i), ladite transformation mettant en œuvre une table de conversion (S) représentant une fonction non
30 linéaire (F), caractérisé par

- des moyens de comparaison (E204, E304, E406) entre une valeur de masquage (A) et ledit premier masque d'entrée;

- des moyens de détermination (E206, E306, E408), à l'aide de ladite table de conversion, d'une donnée de sortie correspondant à la donnée d'entrée masquée à laquelle est appliquée ladite valeur de masquage;

5 - des moyens agencés pour stocker (E206, E306, E408), dans un registre (R_0 , R_1) dépendant du résultat de ladite comparaison, de la donnée de sortie déterminée.

24 - Dispositif selon la revendication 23, comprenant des moyens d'itérations, pour une pluralité de valeurs de masquage, d'une étape de comparaison par lesdits moyens de comparaison, d'une étape de détermination
10 par lesdits moyens de détermination et d'une étape de stockage par lesdits moyens agencés pour stocker

25 - Dispositif selon la revendication précédente, dans lequel les moyens d'itérations sont agencés pour itérer lesdits étapes pour un nombre de valeurs de masquage égale à la taille de la table de conversion de sorte à
15 accéder à chacune des entrées de ladite table de conversion.

26 - Dispositif selon l'une des revendications 23 à 25, dans lequel ladite donnée d'entrée est masquée par un premier et un deuxième masques d'entrée, les moyens de comparaison étant agencés pour comparer ledit premier masque d'entrée et une combinaison entre ladite valeur de masquage
20 et ledit deuxième masque d'entrée.

27 - Dispositif selon la revendication précédente en dépendance de la revendication 24, dans lequel lesdits moyens d'itération sont agencés pour itérer, pour $A=0$ à 2^m-1 , les opérations:

$cmp \leftarrow compare(X_1 \oplus A, X_2)$
25 $R_{cmp} \leftarrow S(M' \oplus A) \oplus Z_1 \oplus Z_2$

puis pour retourner R_0

où X_1 et X_2 sont deux masques d'entrée, m la dimension d'entrée de la table-S, M' la donnée masquée, R_0 et R_1 deux registres, Z_1 et Z_2 deux masques de sortie et *compare* une fonction retournant 0 en cas d'égalité, 1
30 sinon.

28 - Dispositif selon l'une quelconque des revendications 23 à 27, dans lequel les moyens de stockage sont agencés pour additionner ladite donnée déterminée à la valeur dudit registre.

29 - Dispositif selon la revendication précédente en dépendance de la
5 revendication 24, dans lequel les moyens d'itération sont agencés pour retourner, après lesdites itérations, une valeur de sortie formée à partir desdits registres.

30 - Dispositif selon la revendication précédente, dans lequel lesdits
10 moyens d'itération sont agencés pour itérer, pour $A=0$ à 2^m-1 , les opérations suivantes:

$$cmp \leftarrow compare(A, X)$$

$$R_{cmp} \leftarrow R_{cmp} \oplus S(M' \oplus A)$$

puis, postérieurement auxdites itérations, pour comparer lesdits registres et retourner la valeur $R_0 \oplus (cmp \times R_1)$

15 où X est le masque d'entrée, m la dimension d'entrée de la table- S , M' la donnée masquée, R_0 et R_1 deux registres et *compare* une fonction retournant 0 en cas d'égalité, 1 sinon.

31 - Dispositif selon la revendication précédente, comprenant des moyens d'initialisation desdits registres à une valeur de masquage Z .

20 32 - Dispositif selon l'une quelconque des revendications 23 à 31, dans lequel les moyens de comparaison mettent en œuvre une fonction de deux paramètres dont le résultat est un booléen (b) en cas d'égalité entre les deux paramètres et le complémentaire du booléen ($\bar{b}$) sinon, et dans lequel lesdits registres sont indexés par ledit résultat de la fonction de comparaison.

25 33 - Dispositif selon la revendication précédente, comprenant des moyens de génération pseudo-aléatoire dudit booléen (b).

34 - Dispositif selon l'une quelconque des revendications 23 à 33, dans lequel les moyens de comparaison de deux paramètres sont agencés pour lire l'entrée d'une table T à un index fonction de deux paramètres et d'une
30 valeur prédéterminée.

35 - Dispositif selon la revendication précédente, dans lequel les entrées de ladite table T présentent une même valeur booléenne à l'exception

d'une entrée associée à un index ($r3$) pseudo-aléatoire qui présente une valeur complémentaire dudit booléen,

les moyens de comparaison sont agencés pour lire l'entrée $T[(x \oplus r3) \oplus y]$ où x et y sont les deux paramètres et $T[k]$ l'entrée d'index k .

5 36 - Dispositif selon l'une quelconque des revendications 23 à 35, dans lequel lesdits masques sont des masques additifs

37 - Dispositif selon la revendication précédente, dans lequel lesdits masques sont des masques booléens.

10 38 - Dispositif selon l'une quelconque des revendications 23 à 37, comprenant des moyens de masquage de ladite donnée de sortie par au moins un premier masque de sortie.

39 - Dispositif selon l'une quelconque des revendications 23 à 38, dans lequel lesdits registres sont des registres d'un microcontrôleur sécurisé.

15 40 - Dispositif selon la revendication 39, dans lequel la table de conversion est mémorisée en mémoire non-volatile.

41 - Dispositif selon l'une des revendications 23 à 40, dans lequel l'entité électronique est une carte à microcircuit.

42 - Dispositif selon l'une des revendications 23 à 41, dans lequel lesdits moyens de détermination comprennent:

- 20 - un moyen de détermination d'un mot ($S_{machine}(M_H)$) dans une table de conversion en fonction d'une partie de ladite donnée d'entrée (M'_H) et d'une partie correspondante de ladite valeur de masquage (A_H), ledit mot étant composé d'une pluralité de données de sortie de ladite table de conversion et comprenant ladite donnée de sortie;
- 25 - un moyen apte à accéder à ladite donnée de sortie dans ledit mot à partir de l'autre partie de la donnée d'entrée (M'_L) et de l'autre partie de la valeur de masquage (A_L).

43 - Procédé selon la revendication précédente, dans lequel ledit un moyen apte à accéder comprend:

- 30 i) un moyen de séparation dudit mot (U, U') en deux moitiés ($H_0(U), H_1(U), U_H, U_L$) respectivement des bits les plus significatifs et des bits les moins significatifs,

ii) un moyen de sélection d'une desdites moitiés dudit mot (U, U') en fonction des valeurs des bits d'un même index dans respectivement l'autre partie de la donnée d'entrée (M'_L , j') et l'autre partie de la valeur de masquage (A_L , r_j , r_{j1} , r_{j2}).

- 5 44 - Produit programme d'ordinateur comprenant une suite d'instructions aptes, lorsqu'elles sont exécutées par un microprocesseur, à mettre en œuvre un procédé selon l'une des revendications 1 à 22.

1/7

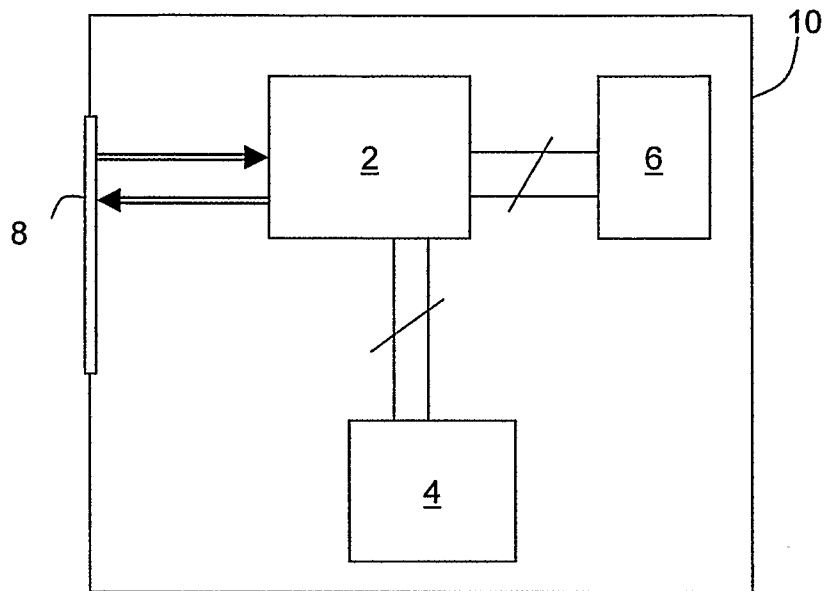


Figure 1

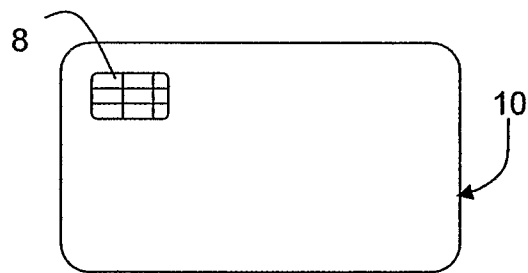
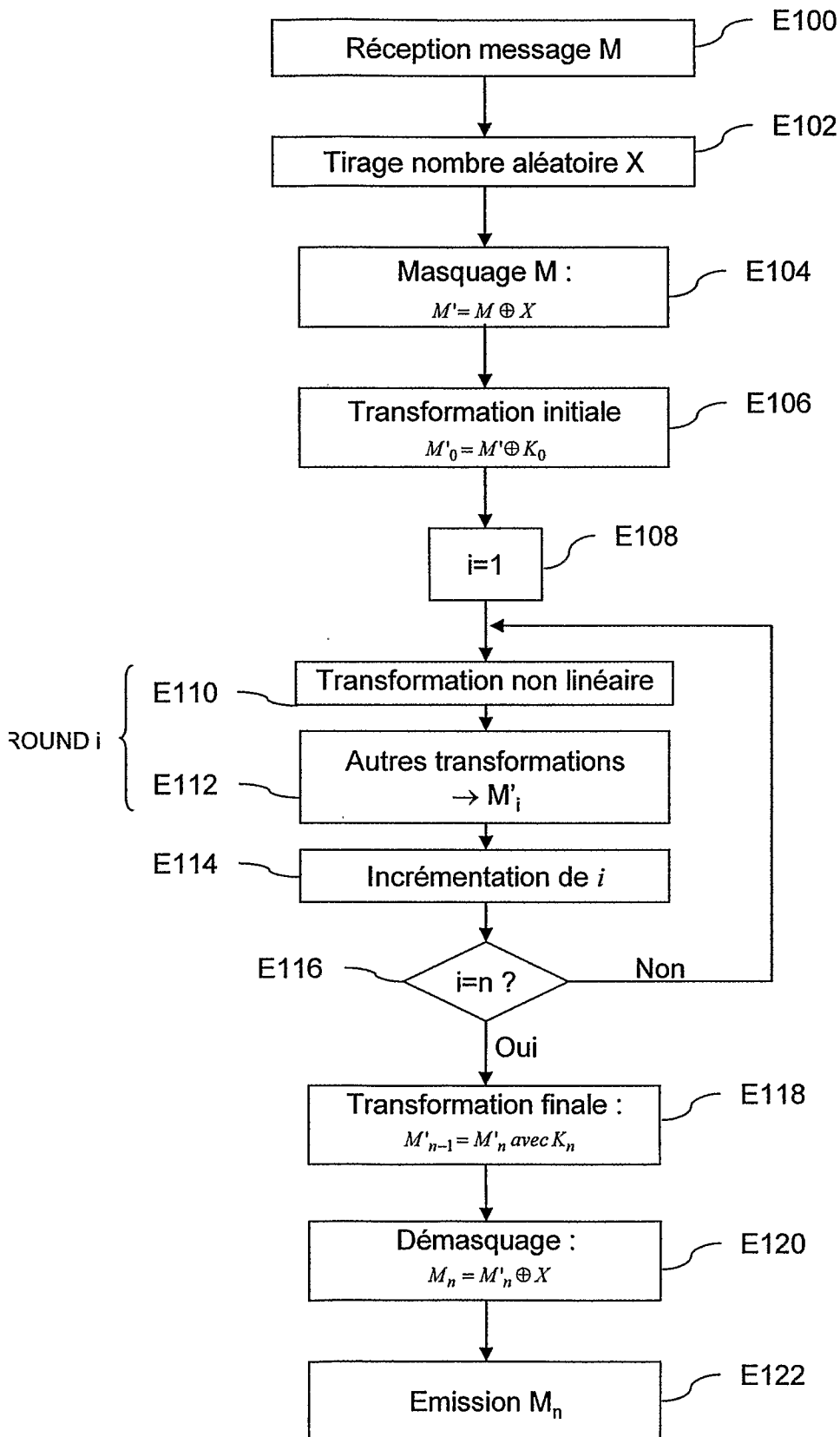
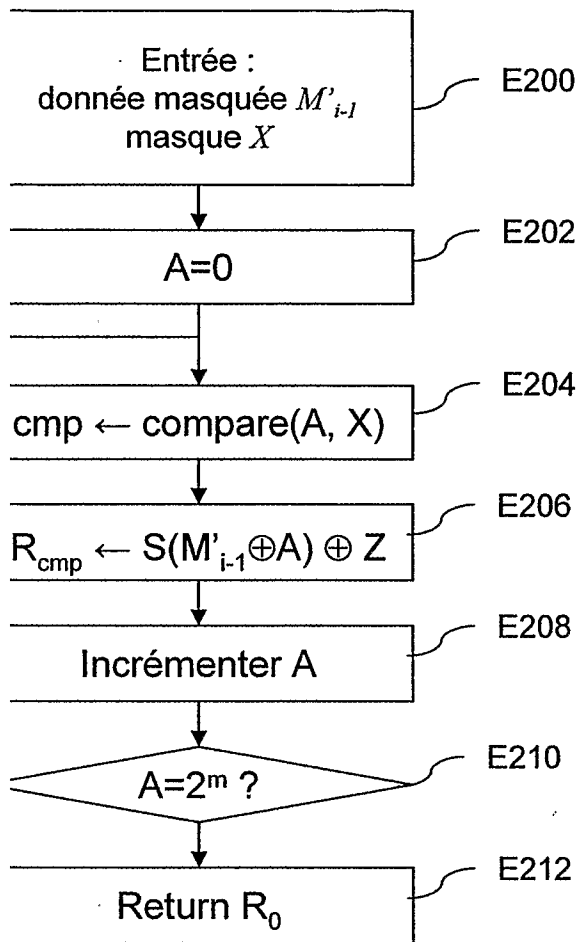
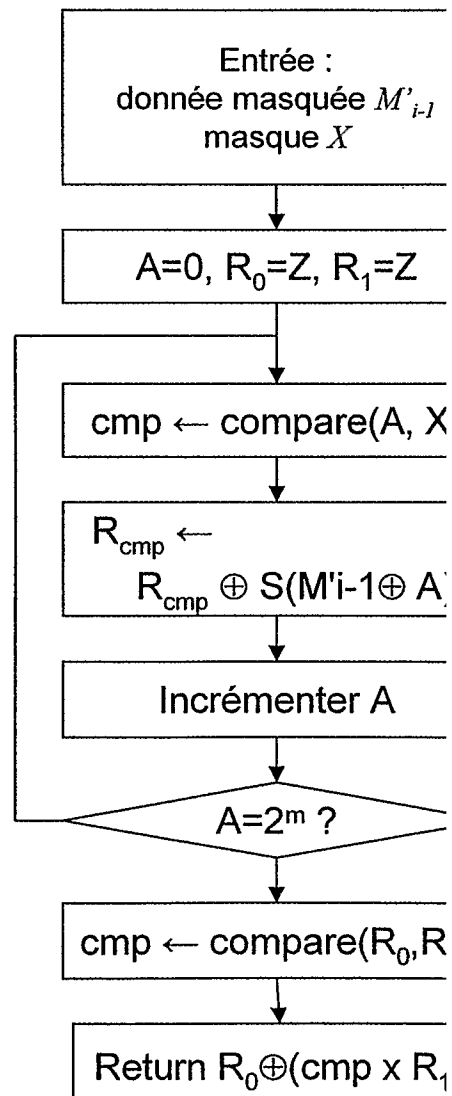


Figure 2

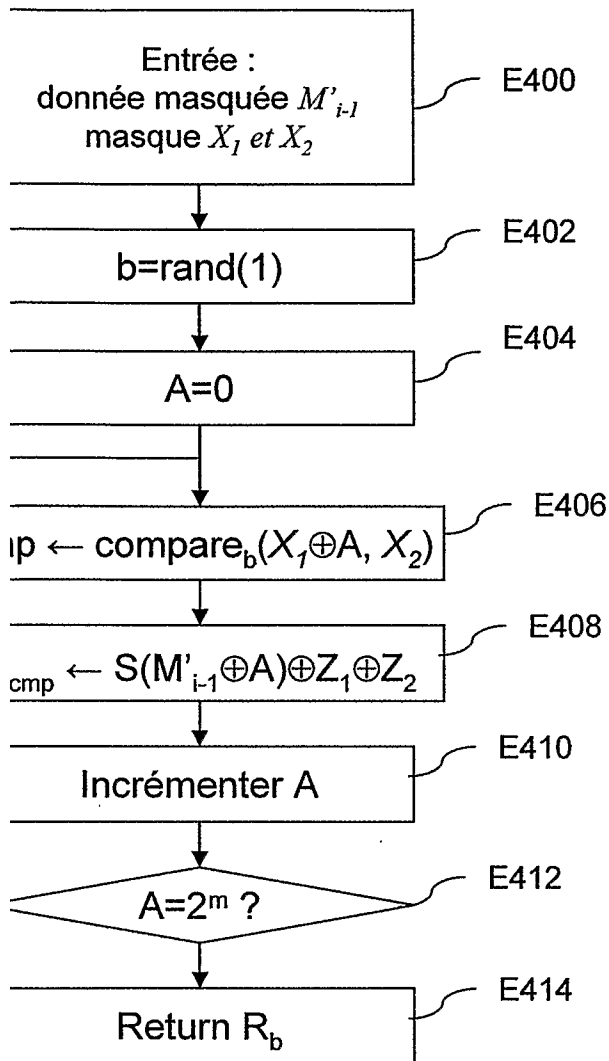
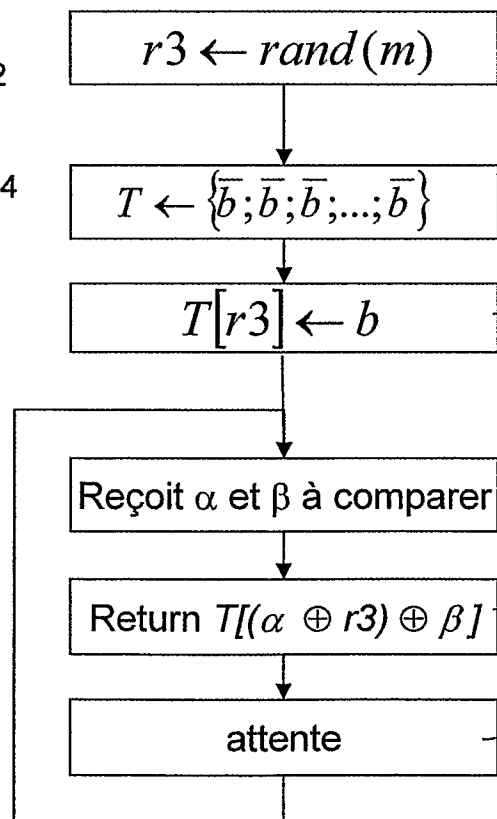
2/7

**Figure**

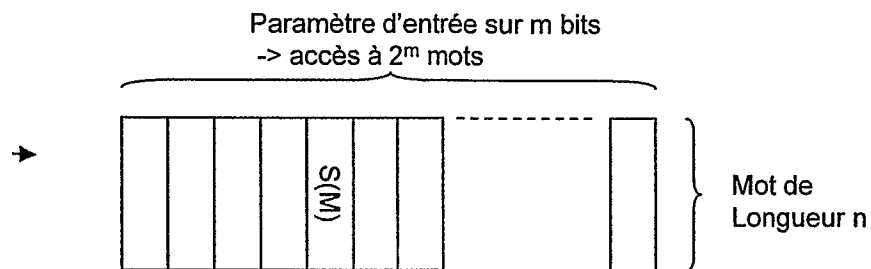
3/7

**Figure 4****Figure 5**

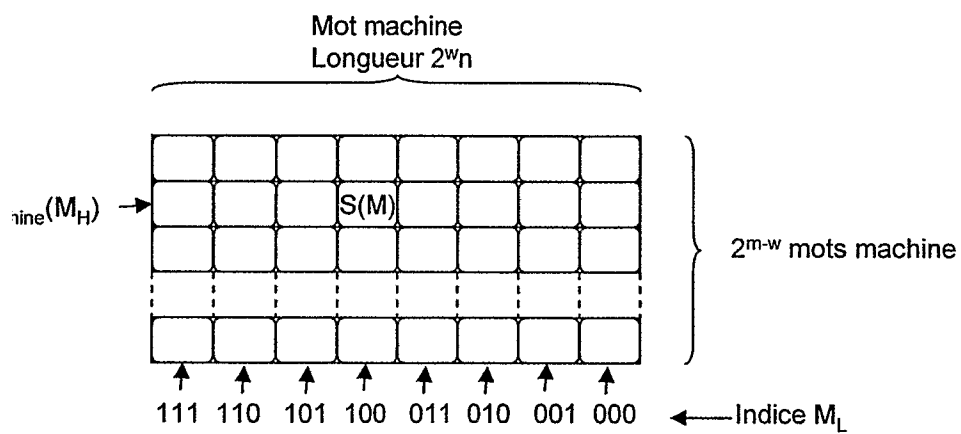
4/7

**Figure 6****Figure 7**

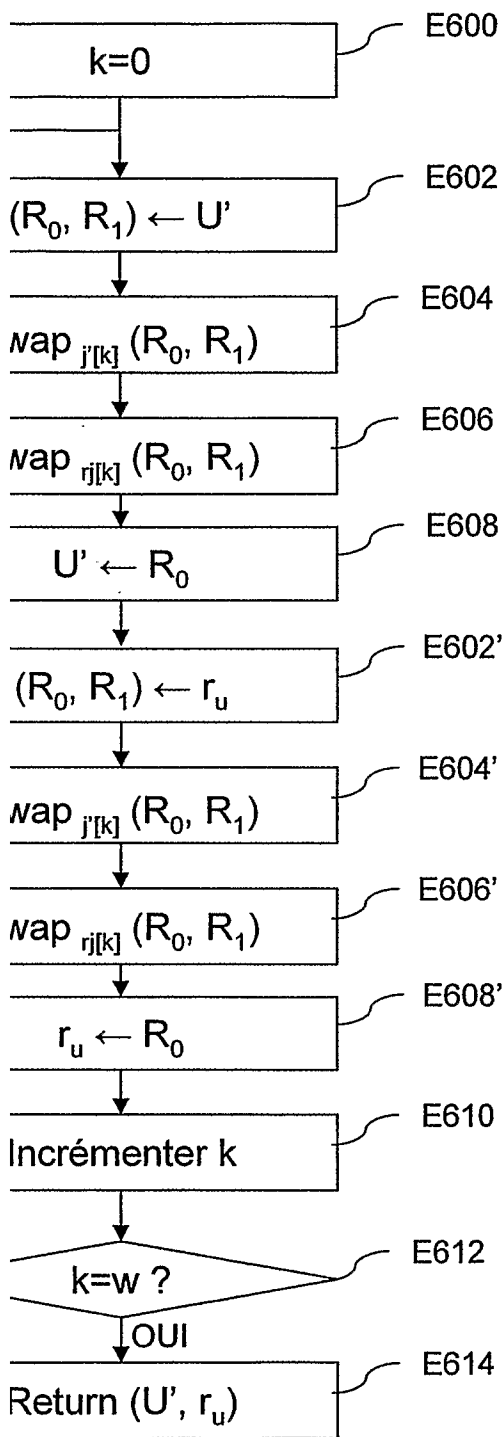
5/7



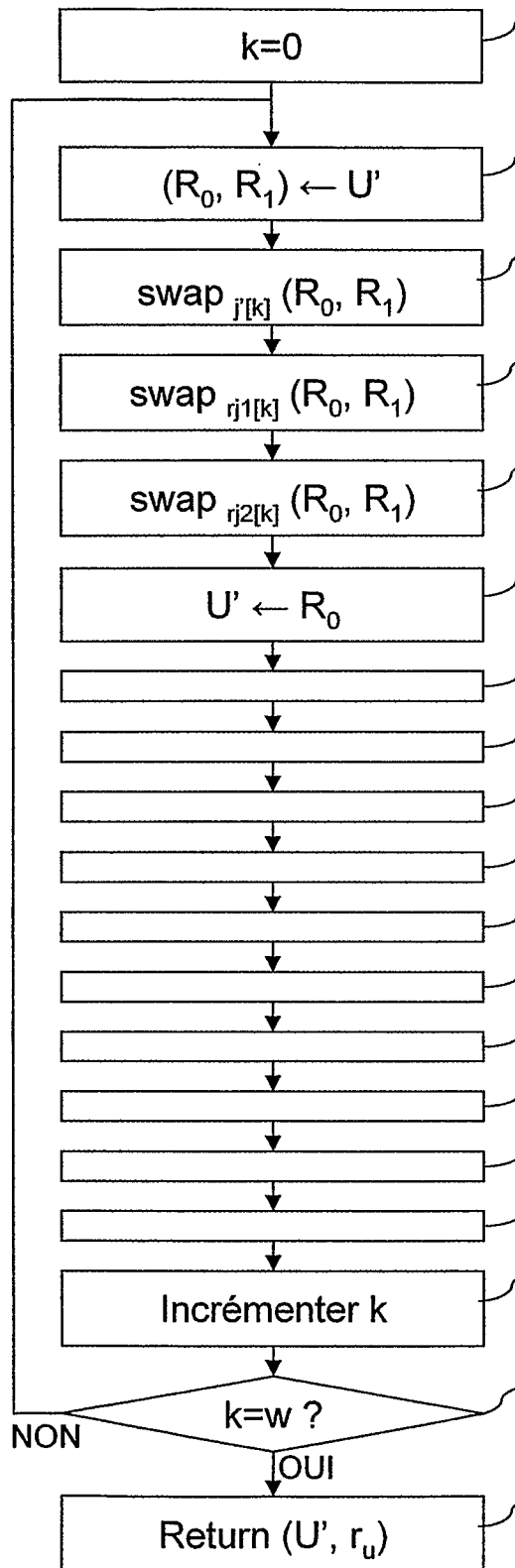
Fig



Fi

**Figure 9**

6/7

**Figure 10**

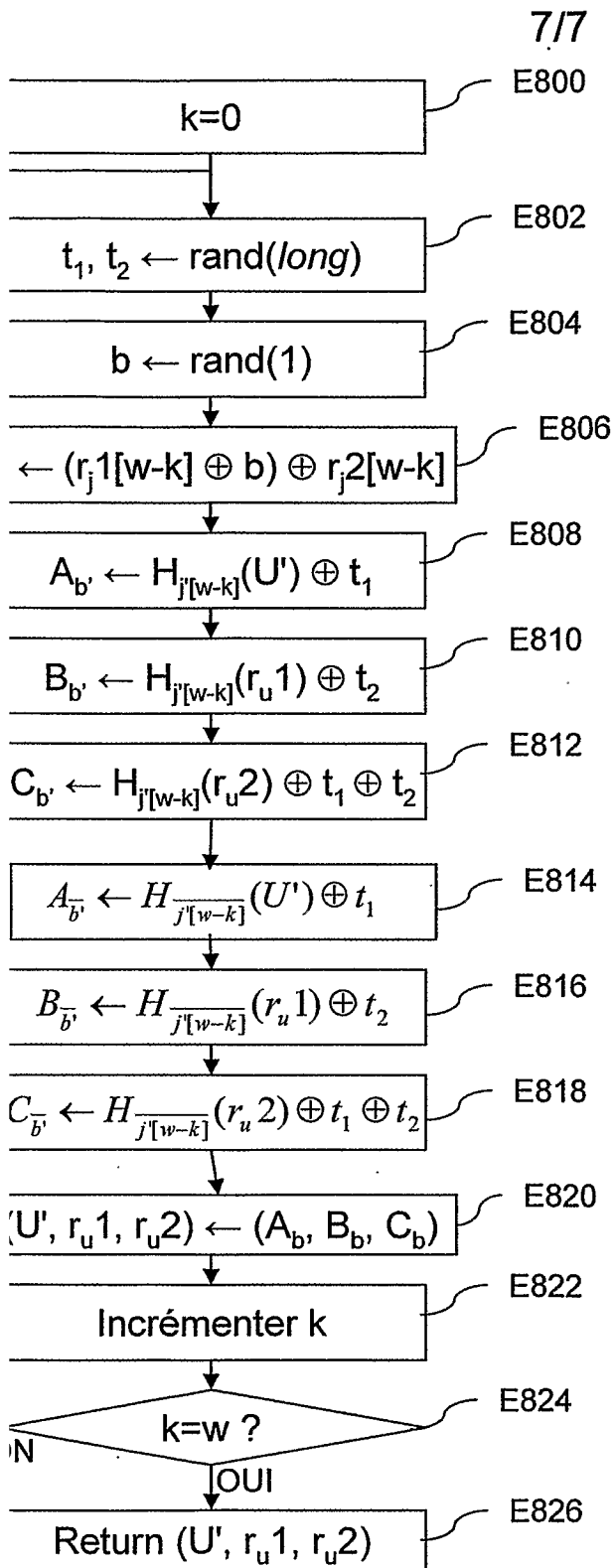


Figure 11

**Déclaration relative à des divulgations non opposables
ou à des exceptions au défaut de nouveauté**

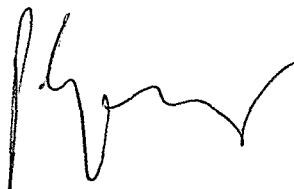
(règles 4.17.v) et 51*bis*.1.a)v))

concernant la demande internationale n° **PCT/FR07/002063**, déposée le
13 décembre 2007 au nom de OBERTHUR CARD SYSTEMS SA.,

Philippe GEYRES, C.E.O déclare qu'un mode de réalisation décrit dans
ladite demande internationale a été divulgué comme suit :

- i) nature de la divulgation : *divulcation réalisée sous forme de publication scientifique à l'occasion de la Conférence WISA (The 8th International Workshop on Information Security Applications), réunion de professionnels techniques du domaine des applications informatiques de sécurité;*
- ii) date de la divulgation : *27-29 août 2007;*
- iii) intitulé de la divulgation : *"A Generic Method for Secure SBox Implementation", auteurs Prouff et Rivain;*
- iv) lieu de la divulgation : *Jeju Island, Corée du Sud.*

Le 5 février 2007,



Philippe GEYRES,
C.E.O

Oberthur Technologies *anciennement Oberthur Card Systems SA*

INTERNATIONAL SEARCH REPORT

International application No
PCT/FR2007/002063

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/06

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)
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, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2007/116140 A (OBERTHUR CARD SYST SA [FR]; GIRAUD CHRISTOPHE [FR]; PROUFF EMMANUEL [F] 18 October 2007 (2007-10-18) cited in the application	1,23,44
A	page 11, line 6 - line 29	2-22, 24-43
A	US 2007/140478 A1 (KOMANO YUICHI [JP] ET AL) 21 June 2007 (2007-06-21) paragraph [0034] - paragraph [0035]	1-44
	-/--	



Further documents are listed in the continuation of Box C.



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* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* 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
- *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.
- *G* document member of the same patent family

Date of the actual completion of the international search

15 octobre 2008

Date of mailing of the international search report

24/10/2008

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

Cretaine, Philippe

INTERNATIONAL SEARCH REPORT

International application No

PCT/FR2007/002063

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	STANDAERT F-X ET AL: "On the Masking Countermeasure and Higher-Order Power Analysis Attacks" INFORMATION TECHNOLOGY: CODING AND COMPUTING, 2005. ITCC 2005. INTERNATIONAL CONFERENCE ON LAS VEGAS, NV, USA 04-06 APRIL 2005, PISCATAWAY, NJ, USA, IEEE, vol. 1, 4 April 2005 (2005-04-04), pages 562-567, XP010795908 ISBN: 978-0-7695-2315-6 page 2	1-44
A	US 2005/259814 A1 (GEBOTYS CATHERINE H [CA]) 24 November 2005 (2005-11-24) paragraph [0041] - paragraph [0043]	1-44

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/FR2007/002063

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2007116140 A	18-10-2007	FR 2899751 A1	12-10-2007
US 2007140478 A1	21-06-2007	JP 2007189659 A	26-07-2007
US 2005259814 A1	24-11-2005	NONE	

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°

PCT/FR2007/002063

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
INV. H04L9/06

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
H04L

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)

EPO-Internal, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	WO 2007/116140 A (OBERTHUR CARD SYST SA [FR]; GIRAUD CHRISTOPHE [FR]; PROUFF EMMANUEL [F] 18 octobre 2007 (2007-10-18) cité dans la demande	1,23,44
A	page 11, ligne 6 - ligne 29	2-22, 24-43
A	US 2007/140478 A1 (KOMANO YUICHI [JP] ET AL) 21 juin 2007 (2007-06-21) alinéa [0034] - alinéa [0035]	1-44
	----- -/--	

☒ Voir la suite du cadre C pour la fin de la liste des documents

☒ Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

A document définissant l'état général de la technique, non considéré comme particulièrement pertinent

E document antérieur, mais publié à la date de dépôt international ou après cette date

L document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

O document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

P document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

T document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention.

X document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

Y document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

Z document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

15 octobre 2008

Date d'expédition du présent rapport de recherche internationale

24/10/2008

Nom et adresse postale de l'administration chargée de la recherche internationale

Office Européen des Brevets, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Cretaine, Philippe

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°

PCT/FR2007/002063

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	STANDAERT F-X ET AL: "On the Masking Countermeasure and Higher-Order Power Analysis Attacks" INFORMATION TECHNOLOGY: CODING AND COMPUTING, 2005. ITCC 2005. INTERNATIONAL CONFERENCE ON LAS VEGAS, NV, USA 04-06 APRIL 2005, PISCATAWAY, NJ, USA, IEEE, vol. 1, 4 avril 2005 (2005-04-04), pages 562-567, XP010795908 ISBN: 978-0-7695-2315-6 page 2	1-44
A	US 2005/259814 A1 (GEBOTYS CATHERINE H [CA]) 24 novembre 2005 (2005-11-24) alinéa [0041] - alinéa [0043]	1-44

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°

PCT/FR2007/002063

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 2007116140 A	18-10-2007	FR 2899751 A1	12-10-2007
US 2007140478 A1	21-06-2007	JP 2007189659 A	26-07-2007
US 2005259814 A1	24-11-2005	AUCUN	