(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
25 juillet 2002 (25.07.2002)

PCT

(10) Numéro de publication internationale
WO 02/058321 A1

(51) Classification internationale des brevets[7] : H04L 9/30, G06F 7/72

(21) Numéro de la demande internationale :
PCT/FR02/00113

(22) Date de dépôt international :
11 janvier 2002 (11.01.2002)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
01/00688 18 janvier 2001 (18.01.2001) FR

(71) Déposant *(pour tous les États désignés sauf US)* : **GEMPLUS** [FR/FR]; Avenue du Pic de Bertagne, Parc d'Activités de Gemenos, F-13420 Gemenos (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants *(pour US seulement)* : **JOYE, Marc** [BE/FR]; 19 rue Voltaire, F-83640 Saint Zacharie (FR). **PAILLIER, Pascal** [FR/FR]; 37 Cours de Vincennes, F-75020 Paris (FR). **CORON, Jean-Sébastien** [FR/FR]; 4 rue Léon Delagrange, F-75015 Paris (FR).

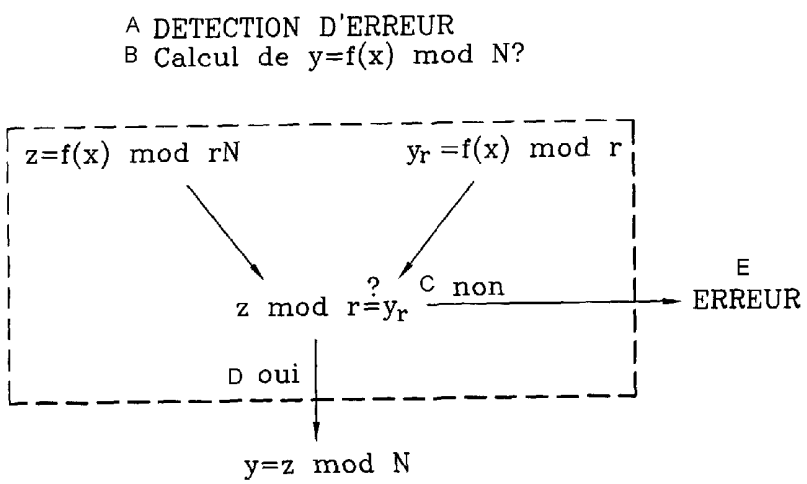
(74) Mandataire : **BRUYERE, Pierre**; Gemplus, BP 100, F-13881 Gemenos Cedex (FR).

(81) États désignés *(national)* : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

[Suite sur la page suivante]

(54) **Title:** DEVICE AND METHOD FOR EXECUTING A CRYPTOGRAPHIC ALGORITHM

(54) **Titre :** DISPOSITIF ET PROCEDE D'EXECUTION D'UN ALGORITHME CRYPTOGRAPHIQUE



(57) **Abstract:** The device for executing a cryptographic algorithm comprises computational means, data storage means and data communication means. The storage means comprise determined values r and N, a predetermined function f(x) with a value x, as well as an algorithm, by means of which the computational means can determine: the value of z = f(x) modulo r.N; the value of y_r = f(x) modulo r; if z modulo r is equal to or not equal to y_r, in order to detect an error in the cryptographic algorithm calculation when z modulo r is not equal to y_r, and to calculate the value y = z modulo N when z modulo r is in fact equal to y_r. One possible field of application for said invention is chip cards.

[Suite sur la page suivante]

WO 02/058321 A1

(84) États désignés *(régional)* **:** brevet ARIPO (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclarations en vertu de la règle 4.17 :

— *relative à l'identité de l'inventeur (règle 4.17.i)) pour les désignations suivantes AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, brevet ARIPO (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *relative au droit du déposant de demander et d'obtenir un brevet (règle 4.17.ii)) pour les désignations suivantes AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, brevet ARIPO (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)*

— *relative au droit du déposant de revendiquer la priorité de la demande antérieure (règle 4.17.iii)) pour toutes les désignations*

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

Publiée :

— *avec rapport de recherche internationale*

— *avant l'expiration du délai prévu pour la modification des revendications, sera republiée si des modifications sont reçues*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : Le dispositif d'exécution d'un algorithme cryptographique comprend des moyens de calcul, des moyens de mémorisation de données et des moyens de communication de données. Les moyens de mémorisation contiennent: des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme permettant aux moyens de calcul d'établir: la valeur de z = f(x) modulo r.N; la valeur de yr = f(x) modulo r; si z modulo r est égal ou pas égal à yr, pour: dans le cas où z modulo r n'est pas égal à yr, constater une erreur dans le calcul de l'algorithme cryptographique, et dans le cas où z modulo r est effectivement égal à yr, calculer la valeur y = z modulo N. Un des domaines d'application de l'invention est dans les cartes à puce.

DISPOSITIF ET PROCEDE D'EXECUTION D'UN ALGORITHME CRYPTOGRAPHIQUE

L'invention concerne le domaine des algorithmes cryptographiques destinés notamment aux dispositifs électroniques communicants, dont un exemple non restrictif est une carte à puce.

Les algorithmes cryptographiques sont couramment exécutés dans de tels dispositifs pour assurer le chiffrement de données émises et/ou le déchiffrement de données reçues lorsque celles-ci doivent demeurer confidentielles. A cette fin, on prévoit un microprocesseur apte à exécuter l'algorithme cryptographique, associé à une mémoire figée (ROM) pour l'enregistrement du programme contenant l'algorithme et une mémoire re-inscriptibe (RAM) pour constituer des registres et contenir les données évolutives. Les informations codées du dispositif transitent entre le microprocesseur et une interface de communication, formant un port vers l'extérieur.

Il est possible pour des fraudeurs d'interférer avec l'algorithme cryptographique en agissant soit au niveau de l'interface de communication, soit au sein du microprocesseur et de ses mémoires, dans le but de casser le code afin de rendre intelligible les données codées ou de modifier ces données à leur avantage.

Pour minimiser ce genre de risque d'attaque, plusieurs stratégies de protection ont déjà été envisagées, tant au niveau de la réalisation matérielle des dispositifs qu'au niveau des processus de calcul.

Dans le domaine de la carte à puce, entre autres, il existe plusieurs attaques possibles, dont une dite "attaque par faute". Dans ce type d'attaque, l'attaquant induit une faute quelconque pendant le

calcul d'un algorithme cryptographique, dans le but d'exploiter la présence de cette faute pour extraire une information secrète.

Ce type d'attaque est notamment envisageable avec l'algorithme RSA (Rivert, Shamir, Adleman), qui est celui le plus utilisé en cryptographie dans ce domaine d'application. La sécurité y est basée sur la factorisation. On établit un nombre N qui est le produit de deux grands nombres premiers p et q, soit N = p.q. Pour signer un nombre x qui exprime un message, on utilise une clé secrète d afin de calculer la valeur $y = x^d$ modulo N. On rappelle de manière générale qu'une valeur v exprimée modulo N (abréviation "v mod N") est égale au reste inférieur à N à l'issue d'une soustraction d'un multiple entier de N; par exemple 11 modulo 3 = 2, soit le reste inférieur à 3 après la soustraction du multiple 3 fois 3.

Pour vérifier que la signature du code est correcte, on utilise une clé correspondante, dite clé publique, qui est un exposant e. On vérifie simplement que $x = y^e$ mod N est égal à la valeur constitutive du message.

La figure 1 illustre le processus de calcul de la signature $y = x^d$ modulo N en utilisant le théorème des restes chinois (TRC). Le théorème des restes chinois est également connu par sa dénomination anglo-saxonne "Chinese remainder theorem" (CRT).

Pour gagner du temps, exécution quatre fois plus rapide de l'algorithme, on n'effectue pas les calculs directement modulo N, mais on effectue d'abord les calculs modulo p et modulo q.

On désigne les valeurs de x modulo p et x modulo q respectivement par x_p et x_q. Par ailleurs, on désigne par d_p la valeur d modulo (p-1), et par d_q la valeur d modulo (q-1).

On effectue le calcul modulo p par calcul de y_p = x_p exposant d_p modulo p. De même, on calcule modulo q la valeur y_q = x_q exposant d_q modulo q.

Après avoir obtenu les valeurs y_p et y_q respectivement modulo p et modulo q, on les recombine par le théorème des restes chinois pour obtenir la valeur y précitée.

Supposons maintenant qu'un attaquant, par une méthode quelconque, induit une erreur durant le calcul de y_p, mais pas durant celui de y_q. Cela impliquerait que la valeur de y_p sera incorrecte. Le fait qu'il s'agisse d'une valeur incorrecte est indiqué par un accent circonflexe au-dessus du "y" dans la figure 1. Par contre, la valeur de y_q sera correcte. De ce fait, lorsque l'on recombinera les valeurs $\hat{y}_p$ et y_q par le TRC, la signature résultante sera incorrecte.

Si l'attaquant connaît la valeur de la clé publique de vérification e, il peut calculer la valeur $\hat{y}^e$ - x modulo N. On a par ailleurs la signature correcte y, égale à x^d modulo N. A partir de la relation préétablie $x = y^e$, l'attaquant n'a seulement à calculer que $\hat{y}^e$ - x modulo N. Il extrait le plus grand commun diviseur (pgcd) avec N, soit : pgdc($\hat{y}^e$ - x mod N,N) = q. Il obtient alors le facteur secret q. De ce fait, le code RSA est effectivement cassé.

Autrement dit, si quelqu'un est capable d'induire une erreur quelconque durant un calcul modulo p alors que le calcul modulo q est correct, il peut casser complètement le code RSA.

Une première contremesure pour éviter ce genre de scénario consiste à recalculer l'ensemble de l'algorithme. On compare les valeurs obtenues à l'issue des calculs successifs. S'ils sont identiques, on suppose qu'il n'y a pas eu d'erreur induite. Un problème avec cette approche est qu'elle ne détecte pas

une faute permanente. Par exemple, on ne pourra déceler une attaque dans laquelle l'erreur induite consiste à forcer systématiquement un bit à un état logique déterminé.

Une autre méthode visant à parer une telle attaque est basée sur une vérification. On obtient une signature qui est calculée par le TRC. Ensuite, on vérifie que la signature est correcte et que la clé publique e est bonne. Cette approche est très fiable, mais l'algorithme de signature ne connaît pas toujours la clé de vérification e, ce qui empêche de pouvoir la mettre en oeuvre dans certaines applications.

Un autre inconvénient de cette méthode est que si e est grand, cela implique deux exponentiations. La signature est alors deux fois plus lente.

Selon une contremesure à l'attaque par faute décrite par Shamir dans le document brevet 5 633 929, on procède par l'algorithme suivant :

1. Choisir un nombre aléatoire r de faible valeur,
2. Calculer :

$$y_{rp} = x^d \bmod rp, \text{ et}$$

$$y_{rq} = x^d \bmod rq ;$$

3. Si $y_{rp} \neq y_{rq} \pmod r$, alors il y a erreur, (peut-être induite par une attaque, et donc interruption de l'algorithme, sinon ;
4. Emettre en sortie : $y = TRC(y_{rp} \bmod p, q_{rq} \bmod q)$.

Ainsi, pour un nombre aléatoire r, au lieu de calculer modulo p, on calcule modulo r.p et modulo r.q. Ensuite, on vérifie que ces deux valeurs sont égales modulo r. Si ces deux valeurs sont différentes, il est certain qu'il y a eu une erreur. Par contre, si elles sont égales, on peut supposer qu'il n'y a pas eu

d'erreur, avec une probabilité de 1/r de se tromper dans cette supposition.

Un inconvénient de cette méthode est que l'on calcule $y_{rp} = x^d \bmod rp$, et non pas $x^{dp} \bmod rp$. Or, la valeur d a la taille du module, qui est généralement un nombre de 1024 bits, tandis que dp est un nombre de la taille de la moitié du module, soit 512 bits dans l'exemple.

Cela implique que dans le schéma normal, sans détection de faute, on effectue une première exponentiation avec un exposant et un module de 512 bits, et une seconde exponentiation avec un exposant et un module de 512 bits. En revanche, avec la méthode de contremesure selon le document brevet 5 633 929, on doit utiliser non pas d_p, mais d. Cela implique que l'exposant aura une taille de 1024 bits de chaque côté. On perd donc en efficacité.

Un autre inconvénient de la méthode Shamir est qu'elle ne fonctionne que pour le mode de calcul basé sur le TRC. Or, il est également envisageable de calculer directement x^d modulo n, c'est-à-dire sans faire appel au théorème des restes chinois.

En effet, il existe deux façons de stocker la clé secrète. Soit on garde la valeur d, soit on garde les valeurs d_p, d_q, p et q. Quand on calcule directement, on utilise le mode standard ; quand on calcule modulo p et modulo q, on utilise le mode TRC.

Au vu de ce qui précède, l'invention propose trois contremesures, notamment aux attaques par défaut, qui autorisent des exponentiations avec un exposant de la taille du module et qui peuvent s'adapter au mode standard ou au mode TRC.

Plus particulièrement, la première contremesure de l'invention concerne, selon un premier objet, un dispositif d'exécution d'un algorithme cryptographique,

comprenant des moyens de calcul, des moyens de mémorisation de données et des moyens de communication de données, caractérisé en ce les moyens de mémorisation contiennent : des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme permettant aux moyens de calcul d'établir :

- la valeur de z = f(x) modulo r.N ;
- la valeur de y_r = f(x) modulo r ;
- si z modulo r est égal ou pas égal à y_r, pour :
  - dans le cas où z modulo r n'est pas égal à y_r, constater une erreur dans le calcul de l'algorithme cryptographique, et
  - dans le cas où z modulo r est effectivement égal à y_r, calculer la valeur y = z modulo N.

Dans un mode de réalisation où l'algorithme est du type à exécution en mode standard, celui-ci permet aux moyens de calcul d'établir:

- la valeur de $z = x^d \bmod rN$;
- si $x^{d \bmod 2 \text{ puissance } s}$ est égal ou pas égal à z modulo (r), pour :
  - dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ n'est pas égal à z modulo (r), constater une erreur dans le calcul de l'algorithme cryptographique, et
  - dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ est égal à z modulo (r), calculer la valeur y = z mod N.

Dans un mode de réalisation où l'algorithme est du type à exécution en mode du théorème des restes chinois (TRC), l'algorithme permet aux moyens de calcul d'établir:

- la valeur de $R_p = r.r_p = r_p \| r_p$ et $K_p = d_p + k_p(p - 1)$, où : r_p est une valeur dans $[0,2^s)$, $k_p \in [0,2^t)$, p est un nombre premier, et d_p = d modulo (p - 1);

- la valeur de $z_p = x^{Kp} \bmod R_p p$;

- le constat d'une erreur dans le calcul de l'algorithme cryptographique si $x^{Kp \bmod 2 \text{ puissance } s}$ n'est pas égal à $z_p \pmod r$;

- la valeur de $R_q = r.r_q = r_q \| r_q$ et $K_q = d_q + k_q(q - 1)$, où : r_q est une valeur dans $[0,2^s)$, $k_q \in [0,2^t)$, q est un nombre premier, et d_q = (d modulo q) - 1;

- la valeur de $z_q = x^{Kq} \bmod R_q q$;

- le constat d'une erreur dans le calcul de l'algorithme cryptographique si $x^{Kq \bmod 2 \text{ puissance } s}$ n'est pas égal à $z_q \pmod r$;

- la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ s'il n'y a pas eu d'erreur constatée.

Dans les modes de réalisation préférés, s = 16, t = 32, la valeur r est un nombre premier, avantageusement égal à $2^{16} + 1$, ou la valeur de r est un multiple d'un nombre premier.

Avantageusement, on établit la valeur N = p.q, où p et q sont des nombres premiers.

La valeur x peut correspondre à des informations protégées, avec la fonction f(x) étant donnée par $f(x) = x^d$, où d est une clé privée.

Avantageusement, la fonction f(x) est égale à g(x)modulo Φ, où $g(x) = x^d$, et Φ est la fonction indicatrice d'Euler du module.

La deuxième contremesure de l'invention s'applique à un mode de réalisation où l'algorithme est du type à exécution en mode du théorème des restes chinois (TRC), l'algorithme permettant aux moyens de calcul d'établir:

- La valeur de $z_p=x\hat{}d_p \bmod p*r$ et la valeur de $z_q=x\hat{}d_q \bmod q*r$ où r est un entier aléatoire;
- La valeur de $b_p=z_p\hat{}d_q \bmod r$ et la valeur de $b_q=z_q\hat{}d_p \bmod r$;
- Le constat d'une erreur dans le calcul si la valeur de $b_p \bmod r$ n'est pas égale à la valeur de $b_q \bmod r$;
- La valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ s'il n'y a pas eu d'erreur constatée.

Dans une variante de l'algorithme ci-dessus appliqué à la deuxième contremesure, il est possible d'utiliser l'entier $d'_p= d_p +r1*(p-1)$ où r1 est un entier aléatoire. Il est aussi possible d'utiliser le résultat du calcul de (x+t*n) à la place de x, où t est un entier aléatoire.

L'avantage de la deuxième contremesure comme de la première contremesure est qu'elles permettent de protéger l'exécution de l'algorithme cryptographique contre les attaques par fautes sur les exponentiations.

La troisième contremesure de l'invention consiste à protéger l'étape de calcul de la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ à partir du procédé suivant :

- Déterminer la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$;
- Vérifier que $(y- z_p)*(y- z_q)=0 \bmod N$. Si ce n'est pas le cas, renvoyer un message d'erreur. Sinon, renvoyer y.

La troisième contremesure permet donc de protéger l'algorithme cryptographique contre une éventuelle attaque par faute sur l'étape du calcul de y par théorème du reste chinois.

Une variante de la troisième contremesure consiste à exécuter le procédé suivant:

- Déterminer la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$;
- Calculer $\alpha=(y- z_p) \bmod p*r$ et $\beta=(y- z_q) \bmod q*r$;

- Soit τ le double de la taille de l'entier r exprimée en nombre de bits;
- Calculer $t=\alpha*\beta/N \bmod 2^{\wedge}\tau$;
- Vérifier que $\alpha.\beta-t*N=0$, le calcul s'effectuant sur les entiers. Si ce n'est pas le cas, renvoyer un message d'erreur. Sinon renvoyer y.

L'avantage de la variante de la troisième contremesure est que les calculs s'effecuent à l'aide d'entiers de taille plus courte que dans la troisième contremesure.

Dans le mode de réalisation, l'algorithme est du type RSA (Rivert, Shamir, Adleman) ; cependant, d'autres types d'algorithmes peuvent être envisagés.

De préférence, le dispositif interrompt la communication de données en cas de constat d'erreur dans lesdits calculs.

Le dispositif en question peut être une carte à puce.

Selon un deuxième objet, l'invention concerne l'utilisation du dispositif précité pour contrer des attaques par défaut.

Selon un troisième objet, l'invention concerne un procédé d'exécution d'un algorithme cryptographique, caractérisé en ce qu'il comprend, à partir de valeurs déterminées r et N, et d'une fonction prédéterminée f(x) d'une valeur x, les étapes de :

- calculer la valeur de z = f(x) modulo r.N ;
- calculer la valeur de y_r = f(x) modulo r ;
- déterminer si z modulo r est égal ou pas égal à y_r, pour :
  - dans le cas où z modulo r n'est pas égal à y_r, constater une erreur dans le calcul de l'algorithme cryptographique, et

- dans le cas où z modulo r est effectivement égal à y_r, calculer la valeur y = z modulo N.

Selon un quatrième objet, l'invention concerne l'utilisation du procédé pour contrer les attaques par faute.

Les caractéristiques optionnelles présentées ci-dessus dans le cadre du dispositif s'appliquent mutatis mutandis à ce procédé.

L'invention et les avantages qui en découlent apparaîtront plus clairement à la lecture de la description qui suit des modes de réalisation préférés, donnés purement à titre d'exemples non-limitatifs, par référence aux dessins annexés dans lesquels :

- la figure 1, déjà analysée, est une représentation symbolique de la méthode de calcul cryptographique de la signature $y = x^d \bmod N$ en utilisant le théorème des restes chinois (TRC) ;

- la figure 2 est un schéma bloc représentant de manière synoptique les éléments d'une carte à puce apte à mettre en oeuvre l'invention ; et

- la figure 3 est une représentation symbolique de l'approche générale à la détection d'erreur dans un algorithme cryptographique conforme à l'invention.

Les modes de réalisation sont décrits dans le cadre de cartes à puce, mais peuvent bien entendu s'appliquer à tous autres dispositifs dotés de moyens de calcul cryptographiques.

Ainsi que le montre la figure 1, la carte à puce 1 comprend un microprocesseur 2 couplé à une mémoire figée (ROM) 4 et à une mémoire vive (RAM) 6, le tout formant un ensemble permettant, entre autres, l'exécution d'algorithmes cryptographiques. Plus précisément, le microprocesseur 2 comporte les moyens de calcul arithmétiques nécessaires à l'algorithme, ainsi que des circuits de transfert de données avec les

mémoires 4 et 6. La mémoire figée 4 contient le programme exécutoire de l'algorithme cryptographique sous forme de code source, alors que la mémoire vive 6 comporte des registres pouvant être mis à jour pour le stockage de résultats des calculs.

La carte à puce 1 comporte aussi une interface de communication 8 reliée au microprocesseur 2 pour permettre l'échange de données avec l'environnement extérieur. L'interface de communication 8 peut être du type "à contacts", étant dans ce cas formée d'un ensemble de plots de contacts destinés à se connecter à un contacteur d'un dispositif externe, tel qu'un lecteur de cartes, et/ou du type "sans contact". Dans ce dernier cas, l'interface de communication 8 comporte une antenne et des circuits de communication par voie hertzienne permettant un transfert de données par liaison sans fil. Cette liaison peut aussi permettre un transfert d'énergie d'alimentation des circuits de la carte 1.

L'ensemble des moyens matériels constitutifs de la carte sont connus et ne seront pas décrits de manière détaillée par souci de concision.

Dans l'exemple, l'algorithme cryptographique est du type RSA (de Rivert, Shamir, Adleman), dont les caractéristiques ont été décrites dans la partie introductive.

On s'intéressera dans ce qui suit plus particulièrement à la détection d'erreurs dans le calcul algorithmique et des contremesures conformes à la présente invention. L'erreur en question peut être provoquée délibérément par un attaquant qui vise à casser le code cryptographique utilisé par la carte à puce, comme expliqué dans la partie introductive. Ainsi, pour faire face à cette éventualité, les

contremesures permettent de déceler de telles erreurs et de réagir en conséquence.

Le principe de détection d'erreur est représenté schématiquement à la figure 3. De façon générale, l'exécution de l'algorithme cryptographique implique un calcul d'une fonction f(x) modulo N, quelle que soit la fonction f(x). Ainsi, pour le cas d'un algorithme RSA, on prend une valeur de N qui est le produit de deux grands nombres premiers p et q.

Ainsi que le montre la figure 3, on prend un nombre aléatoire r, et on calcule d'une part z = f(x) modulo rN, et y_r = f(x) modulo r. Ensuite on vérifie que z mod r = y_r. Si cela n'est pas le cas, on est certain qu'il y a une erreur; sinon on suppose, avec une probabilité de se tromper de 1/r, qu'il n'y a pas d'erreur.

Ensuite, pour retrouver la valeur de y = f(x) modulo N, on calcule simplement y = z mod N.

Dans l'implémentation, r est un nombre de 32 bits.

Il sera maintenant décrit comment appliquer une contremesure conforme à l'invention lorsque l'algorithme est exécuté en mode standard. En mode standard, on calcule de façon "brutale" la valeur de x^d mod N, où d est un nombre qui constitue une clé secrète du code.

De manière générale, on procède comme suit. On calcule la valeur d'une part de f(x) modulo rN, et d'autre part de f(x) modulo r ; on vérifie que ces deux valeurs calculées sont égales. Si tel est le cas, on suppose qu'il n'y a pas d'erreur.

Lorsque l'on applique ceci à l'algorithme RSA, on procède comme suit. On ne prend plus un nombre aléatoire, mais un nombre déterminé, qui est ici le nombre $2^{16}+1$. Ce nombre a la propriété intéressante d'être un nombre premier. On calcule simplement la

valeur $z = x^d$ modulo $(2^{16}+1).N$. Ensuite, on calcule la valeur x^d mod $(2^{16} + 1)$.

Dans le mode de réalisation, on ne calcule pas la valeur de x^d, mais on calcule plutôt la valeur de $x^{d \bmod \Phi}$ mod $(2^{16} + 1)$, où Φ est la fonction indicatrice d'Euler du module. On peut ainsi réduire d modulo Φ $(2^{16} + 1)$.

On note que quand un nombre prend comme valeur un nombre premier P, on a la condition: $\Phi(P) = P-1$. Appliquée à l'exemple, cette condition donne : $\Phi(2^{16} + 1) = 2^{16}$.

De ce fait, on n'a pas à calculer x^d modulo $(2^{16} + 1)$, mais plutôt $x^{d \bmod 2 \text{ puissance } 16}$ modulo $(2^{16} + 1)$. Cette valeur est un nombre de 16 bits seulement. L'opération est donc très rapide.

Un autre avantage de cette façon de procéder est que d modulo 2^{16} est facile à calculer, s'agissant des 16 derniers bits de la clé secrète d.

Dans le mode standard, on ne va plus prendre un nombre aléatoire, mais un nombre premier, ou un nombre premier multiplié par un nombre aléatoire. La vérification va chaque fois se faire modulo le nombre premier que l'on a choisi. En procédant de cette manière, on peut réduire le temps de calcul. En effet, la fonction Φ indicatrice d'Euler ne peut être évaluée facilement que pour les nombres premiers.

Par contre - et c'est là où réside la force de l'algorithme RSA - pour casser le code RSA modulo N (= p.q), on doit calculer $\Phi(N)$. La valeur de cette fonction est égale à (p-1).(q-1). Si on ne connaît pas la factorisation de N, on ne peut pas calculer $\Phi(N)$.

Ainsi, la contremesure conforme à l'exemple pour le mode standard revient à effectuer l'algorithme suivant:

1. Calculer $z = x^d \bmod (2^{16} + 1)N$ {pas de nombre aléatoire utilisé, mais une fonction Φ indicatrice d'Euler} ;

2. Si $x^{d \bmod 2 \text{ puissance } 16} \neq z \pmod{(2^{16} + 1)}$ {d mod 2^{16} correspond aux 16 bits de poids faible de d}, alors émettre en sortie ERREUR and cesser l'algorithme, sinon;

3. Emettre en sortie $y = z \bmod N$.

On comprend qu'il est ainsi possible de déceler une faute ayant pour origine possible une attaque, et donc de prendre des mesures préventives. Celles-ci consistent notamment à stopper le processus algorithmique et d'interrompre tout échange de données avec l'interface de communication 8.

Il sera maintenant décrit comment appliquer une contremesure conforme à l'invention lorsque l'algorithme est exécuté en mode de calcul basé sur le théorème des restes chinois (TRC), désigné ci-après mode TRC.

En mode TRC, on réalise simplement les calculs modulo p et modulo q.

On peut effectuer des calculs sur la base d'un modulo d'un nombre premier multiplié par le module (N), ou on peut prendre un nombre premier multiplié par un nombre aléatoire. On fera toujours la vérification sur la base d'un modulo d'un nombre premier. On produit ainsi un calcul avec un modulo d'un premier - 1.

A titre d'exemple, on considèrera, en mode TRC, le calcul modulo p. On choisit un nombre aléatoire k, par exemple de 16 bits. On va l'utiliser pour contrer d'autres attaques. On va aussi prendre k_p, un nombre

de 32 bits, pour éviter d'autres attaques, telles que des attaques en courant ou autres. On rappellera qu'une attaque en courant est basée sur l'analyse du courant consommé par le processeur à diverses étapes du calcul, visant à déterminer par exemple, les caractéristiques d'un calcul d'exponentiation en cours.

On établit la relation suivante : $2^{16}+1$ multiplié par une valeur r_p, est égal à cette valeur r_p que l'on concatène avec lui-même (ici, r_p est une valeur de 16 bits). Cela vaut pour n'importe quel nombre premier. On calcule une valeur K_p (pour rendre l'exposant aléatoire).

On calcule la valeur $z_p = x^{Kp}$ modulo $R_p.p$.

Ensuite, on vérifie que les calculs modulo $2^{16}+1$ sont égaux. Si tel est le cas, on peut supposer, avec un risque de se tromper de $1/2^{16}$, que l'on n'aura pas d'erreur.

Ce qui précède reflète le principe général. Pour l'algorithme cryptographique RSA, on ne va pas prendre un nombre aléatoire, mais un nombre premier. De cette manière, on peut réduire l'exposant modulo un nombre premier - 1. On peut aussi prendre un nombre quelconque multiplié par un nombre premier.

En résumé, la contremesure conforme à l'exemple pour le mode TRC revient à effectuer l'algorithme suivant:

1. Choisir de manière aléatoire r_p dans $[0,2^{16})$ et $k_p \in [0,2^{32})$ (pour parer contre les attaques en courant);

2. Soit $R_p = (2^{16} + 1)r_p = r_p \| r_p$ et $K_p = d_p + k_p(p - 1)$;

3. calculer $z_p = x^{Kp} \bmod R_p p$;

4. Si $x^{Kp \bmod 2 \text{ puissance } 16} \neq z_p \pmod{(2^{16} + 1)}$ alors émettre en sortie ERREUR and cesser l'algorithme, sinon;

5. Répéter les opérations 1 à 4 modulo q ;

6. Emettre en sortie $y = TRC(z_p \bmod p, z_q \bmod q)$.

(Le symbole || indique une concaténation; ainsi a||b = la concaténation de a et de b. Par exemple, pour a = 1011 et b = 1101, alors a||b = 10111101.)

On note de ce qui précède que l'on réalise ces opérations pour un R qui est soit un nombre premier, soit un nombre premier multiplié par un nombre quelconque. La vérification se fait toujours modulo ce nombre premier, de façon plus générale, une puissance première.

L'invention est valable non seulement pour les algorithmes cryptographiques RSA, présentés ici uniquement à titre d'illustration, mais pour tous les algorithmes cryptographiques où l'on travaille en arithmétique modulaire, puisque cette technique permet de vérifier que n'importe quelle fonction modulaire est correcte ou pas.

REVENDICATIONS

1. Dispositif (1) d'exécution d'un algorithme cryptographique, comprenant des moyens de calcul (2), des moyens de mémorisation de données (4, 6) et des moyens de communication de données (8), caractérisé en ce les moyens de mémorisation (4,6) contiennent : des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme permettant aux moyens de calcul (2) d'établir :

- la valeur de z = f(x) modulo r.N ;
- la valeur de y_r = f(x) modulo r ;
- si z modulo r est égal ou pas égal à y_r, pour :

- dans le cas où z modulo r n'est pas égal à y_r, constater une erreur dans le calcul de l'algorithme cryptographique, et
- dans le cas où z modulo r est effectivement égal à y_r, calculer la valeur y = z modulo N.

2. Dispositif selon la revendication 1, caractérisé en ce que l'algorithme est du type à exécution en mode standard, et permet aux moyens de calcul (2) d'établir:

- la valeur de $z = x^d \bmod rN$;
- si $x^{d \bmod 2 \text{ puissance } s}$ est égal ou pas égal à z modulo (r), pour :
  - dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ n'est pas égal à z modulo (r), constater une erreur dans le calcul de l'algorithme cryptographique, et
  - dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ est égal à z modulo (r), calculer la valeur y = z mod N.

3. Dispositif selon la revendication 1, caractérisé en ce que l'algorithme est du type à exécution en mode du théorème des restes chinois (TRC), et permet aux moyens de calcul (2) d'établir:

- la valeur de $R_p = r.r_p = r_p \| r_p$ et $K_p = d_p + k_p(p - 1)$, où : r_p est une valeur dans $[0,2^s)$, $k_p \in [0,2^t)$, p est un nombre premier, et d_p = d modulo (p - 1);
- la valeur de $z_p = x^{Kp}$ mod $R_p p$;
- le constat d'une erreur dans le calcul de l'algorithme cryptographique si $x^{Kp \text{ mod } 2 \text{ puissance } s}$ n'est pas égal à z_p (mod r) ;
- la valeur de $R_q = r.r_q = r_q \| r_q$ et $K_q = d_q + k_q(q - 1)$, où : r_q est une valeur dans $[0,2^s)$, $k_q \in [0,2^t)$, q est un nombre premier, et d_q = (d modulo q) - 1;
- la valeur de $z_q = x^{Kq}$ mod $R_q q$;
- le constat d'une erreur dans le calcul de l'algorithme cryptographique si $x^{Kq \text{ mod } 2 \text{ puissance } s}$ n'est pas égal à z_q (mod r) ;
- la valeur y = TRC(z_p mod p, z_q mod q) s'il n'y a pas eu d'erreur constatée.

4. Dispositif selon la revendication 2 ou 3, caractérisé en ce que s = 16.

5. Dispositif selon la revendication 3 ou 4, caractérisé en ce que t = 32.

6. Dispositif (1) d'exécution d'un algorithme cryptographique, comprenant des moyens de calcul (2), des moyens de mémorisation de données (4, 6) et des moyens de communication de données (8), caractérisé en ce les moyens de mémorisation (4,6) contiennent : des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme du type à exécution en mode du théorème des restes chinois (TRC) permettant aux moyens de calcul (2) d'établir :
- La valeur de z_p=x^d_p mod p*r et la valeur de z_q=x^d_q mod q*r où r est un entier aléatoire;

- La valeur de $b_p=z_p\hat{}d_q \bmod r$ et la valeur de $b_q=z_q\hat{}d_p \bmod r$;
- Le constat d'une erreur dans le calcul si la valeur de $b_p \bmod r$ n'est pas égale à la valeur de $b_q \bmod r$;
- La valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ s'il n'y a pas eu d'erreur constatée.

7. Dispositif d'exécution d'un algorithme cryptographique selon la revendication 6, caractérisé en ce que l'entier $d'_p= d_p +r1*(p-1)$ est utilisé à la place de l'entier d_p ,où r1 est un entier aléatoire.

8. Dispositif d'exécution d'un algorithme cryptographique selon la revendication 6, caractérisé en ce que l' entier (x+t*n) est utilisé à la place de x, où t est un entier aléatoire.

9. Dispositif (1) d'exécution d'un algorithme cryptographique, comprenant des moyens de calcul (2), des moyens de mémorisation de données (4, 6) et des moyens de communication de données (8), caractérisé en ce les moyens de mémorisation (4,6) contiennent : des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme du type à exécution en mode du théorème des restes chinois (TRC) permettant aux moyens de calcul (2) d'établir :
- la valeur de $y = TRC(z_p \bmod p, z_q \bmod q)$;
- le constat d'une erreur de calcul si $(y- z_p)*(y- z_q)$ est différent de 0 modulo N. Dans ce cas, la valeur de y n'est pas renvoyée.

10. Dispositif d'exécution d'un algorithme cryptographique suivant la revendication 9, permettant aux moyens de calcul d'établir :

- la valeur de $\alpha=(y- z_p) \bmod p*r$ et $\beta=(y- z_q) \bmod q*r$
- la valeur de τ qui est le double de la taille de l'entier r exprimée en nombre de bits.
- la valeur de $t=\alpha*\beta/N \bmod 2^{\wedge}\tau$.
- le constat d'une erreur de calcul si $\alpha.\beta-t*N$ est différent de 0. Dans ce cas, la valeur de y n'est pas renvoyée.

11. Dispositif selon l'une quelconque des revendications 1 à 10, caractérisé en ce que la valeur r est un nombre premier.

12. Dispositif selon la revendication 11, caractérisé en ce que r est égal à $2^{16} + 1$.

13. Dispositif selon la revendication 11, caractérisé en ce que la valeur de r est un multiple d'un nombre premier.

14. Dispositif selon l'une quelconque des revendications 1 à 13, caractérisé en ce que la valeur N = p.q, où p et q sont des nombres premiers.

15. Dispositif selon l'une quelconque des revendications 1 à 14, caractérisé en ce que la valeur x correspond à des informations protégées, la fonction f(x) étant donnée par $f(x) = x^d$, où d est une clé privée.

16. Dispositif selon l'une quelconque des revendications 1 à 15, caractérisé en ce que la fonction f(x) est égale à g(x)modulo Φ, où $g(x) = x^d$, et Φ est la fonction indicatrice d'Euler du module.

17. Dispositif selon l'une quelconque des revendications 1 à 16, caractérisé en ce que l'algorithme est du type RSA (Rivert, Shamir, Adleman).

18. Dispositif selon l'une quelconque des revendications 1 à 17, caractérisé en ce qu'il interrompt la communication de données en cas de constat d'erreur dans lesdits calculs.

19. Dispositif selon l'une quelconque des revendications 1 à 18, caractérisé en ce qu'il s'agit d'une carte à puce (1).

20. Utilisation du dispositif selon l'une quelconque des revendications 1 à 19 pour contrer des attaques par faute.

21. Procédé d'exécution d'un algorithme cryptographique, caractérisé en ce qu'il comprend, à partir de valeurs déterminées r et N, et d'une fonction prédéterminée f(x) d'une valeur x, les étapes de :

- calculer la valeur de z = f(x) modulo r.N ;
- calculer la valeur de y_r = f(x) modulo r ;
- déterminer si z modulo r est égal ou pas égal à y_r, pour :
  - dans le cas où z modulo r n'est pas égal à y_r, constater une erreur dans le calcul de l'algorithme cryptographique, et
  - dans le cas où z modulo r est effectivement égal à y_r, calculer la valeur y = z modulo N.

22. Procédé selon la revendication 21, caractérisé en ce que l'algorithme est du type à exécution en mode standard, et comprend les étapes de :

- calculer la valeur de $z = x^d \bmod rN$;

- déterminer si $x^{d \bmod 2 \text{ puissance } s}$ est égal ou pas égal à z modulo (r), pour :

- dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ n'est pas égal à z modulo (r), constater une erreur dans le calcul de l'algorithme cryptographique, et

- dans le cas où $x^{d \bmod 2 \text{ puissance } s}$ est égal à z modulo (r), calculer la valeur y = z mod N.

23. Procédé selon la revendication 21, caractérisé en ce que l'algorithme est du type à exécution en mode du théorème des restes chinois (TRC), et comprend les étapes de :

- calculer la valeur de $R_p = r.r_p = r_p||r_p$ et $K_p = d_p + k_p(p - 1)$, où : r_p est une valeur dans $[0,2^s)$, $k_p \in [0,2^t)$, p est un nombre premier, et d_p = (d modulo p) - 1;

- calculer la valeur de $z_p = x^{Kp} \bmod R_p p$;

- constater une erreur dans le calcul de l'algorithme cryptographique si $x^{Kp \bmod 2 \text{ puissance } s}$ n'est pas égal à $z_p \pmod r$;

- calculer la valeur de $R_q = r.r_q = r_q||r_q$ et $K_q = d_q + k_q(q - 1)$, où : r_q est une valeur dans $[0,2^s)$, $k_q \in [0,2^t)$, q est un nombre premier, et d_q = d modulo (q - 1);

- calculer la valeur de $z_q = x^{Kq} \bmod R_q q$;

- constater une erreur dans le calcul de l'algorithme cryptographique si $x^{Kq \bmod 2 \text{ puissance } s}$ n'est pas égal à $z_q \pmod r$; et

- calculer la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ s'il n'y a pas eu d'erreur constatée.

24. Procédé selon la revendication 22 ou 23, caractérisé en ce que s = 16.

25. Procédé selon la revendication 22 ou 23, caractérisé en ce que t = 32.

26. Procédé d'exécution d'un algorithme cryptographique, caractérisé en ce qu'il comprend, à partir de valeurs déterminées r et N, et d'une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un algorithme du type à exécution en mode du théorème des restes chinois (TRC), les étapes de calculs suivantes :

- calculer la valeur de $z_p=x^{d_p} \bmod p*r$ et la valeur de $z_q=x^{d_q} \bmod q*r$ où r est un entier aléatoire;
- calculer la valeur de $b_p=z_p^{d_q} \bmod r$ et la valeur de $b_q=z_q^{d_p} \bmod r$;
- déterminer le constat d'une erreur dans le calcul si la valeur de $b_p \bmod r$ n'est pas égale à la valeur de $b_q \bmod r$;
- calculer la valeur $y = TRC(z_p \bmod p, z_q \bmod q)$ s'il n'y a pas eu d'erreur constatée.

27. Procédé d'exécution d'un algorithme cryptographique selon la revendication 26, caractérisé en ce que le calcul de l'entier $d'_p= d_p +r1*(p-1)$ est utilisé à la place de l'entier d_p ,où r1 est un entier aléatoire.

28. Procédé d'exécution d'un algorithme cryptographique selon la revendication 26, caractérisé en ce que l' entier (x+t*n) est utilisé à la place de x, où t est un entier aléatoire.

29. Procédé(1) d'exécution d'un algorithme cryptographique, caractérisé en ce qu'il comprend, à partir des valeurs déterminées r et N, une fonction prédéterminée f(x) d'une valeur x, ainsi qu'un

algorithme du type à exécution en mode du théorème des restes chinois (TRC), les étapes de calculs suivantes :

- calculer la valeur de $y = TRC(z_p \bmod p, z_q \bmod q)$;
- déterminer le constat d'une erreur de calcul si $(y- z_p)*(y- z_q)$ est différent de 0 modulo N et dans ce cas, la valeur de y n'est pas renvoyée.

30. Procédé d'exécution d'un algorithme cryptographique selon la revendication 29, caractérisé en ce qu'il comprend les étapes suivantes :

- calculer la valeur de $\alpha=(y- z_p) \bmod p*r$ et $\beta=(y- z_q) \bmod q*r$
- déterminer la valeur de τ qui est le double de la taille de l'entier r exprimée en nombre de bits.
- calculer la valeur de $t=\alpha*\beta/N \bmod 2^{\wedge}\tau$.
- déterminer le constat d'une erreur de calcul si $\alpha.\beta- t*N$ est différent de 0 et dans ce cas, la valeur de y n'est pas renvoyée.

31. Procédé selon l'une quelconque des revendications 21 à 30, caractérisé en ce que la valeur r est un nombre premier.

32. Procédé selon la revendication 31, caractérisé en ce que r est égal à $2^{16} + 1$.

33. Procédé selon la revendication 32, caractérisé en ce que la valeur de r est un multiple d'un nombre premier.

34. Procédé selon l'une quelconque des revendications 21 à 33, caractérisé en ce que la valeur $N = p.q$, où p et q sont des nombres premiers.

35. Procédé selon l'une quelconque des revendications 21 à 34, caractérisé en ce que la valeur x correspond à des informations protégées, la fonction f(x) étant donnée par $f(x) = x^d$, où d est une clé privée.

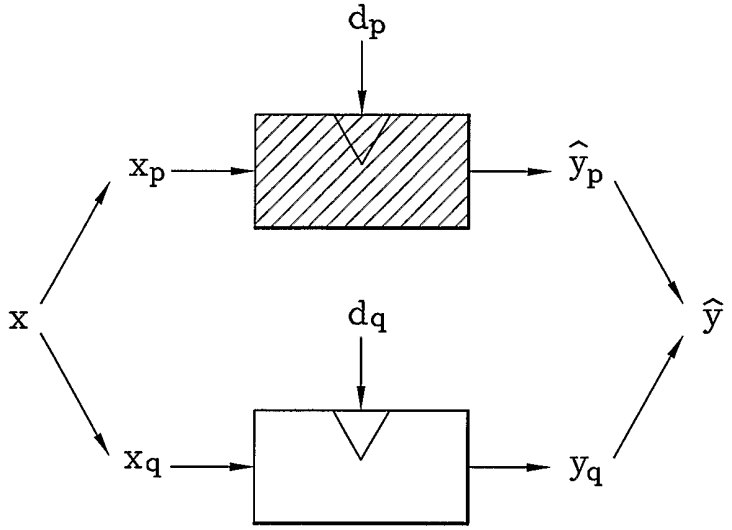
36. Procédé selon l'une quelconque des revendications 21 à 35, caractérisé en ce que la fonction f(x) est égale à g(x)modulo Φ, où $g(x) = x^d$, et Φ est la fonction indicatrice d'Euler du module.

37. Procédé selon l'une quelconque des revendications 21 à 36, caractérisé en ce que l'algorithme est du type RSA (Rivert, Shamir, Adleman).

38. Procédé selon l'une quelconque des revendications 21 à 37, caractérisé en ce qu'il s'interrompt en cas en cas de constat d'erreur dans lesdits calculs.

39. Utilisation du procédé selon l'une quelconque des revendications 21 à 38 pour contrer des attaques par faute.

calcul de signature $y=x^d \bmod N$ utilisant le TRC



$\text{pgcd}(\hat{y}^e - x \bmod N, N) = q$

Fig. 1

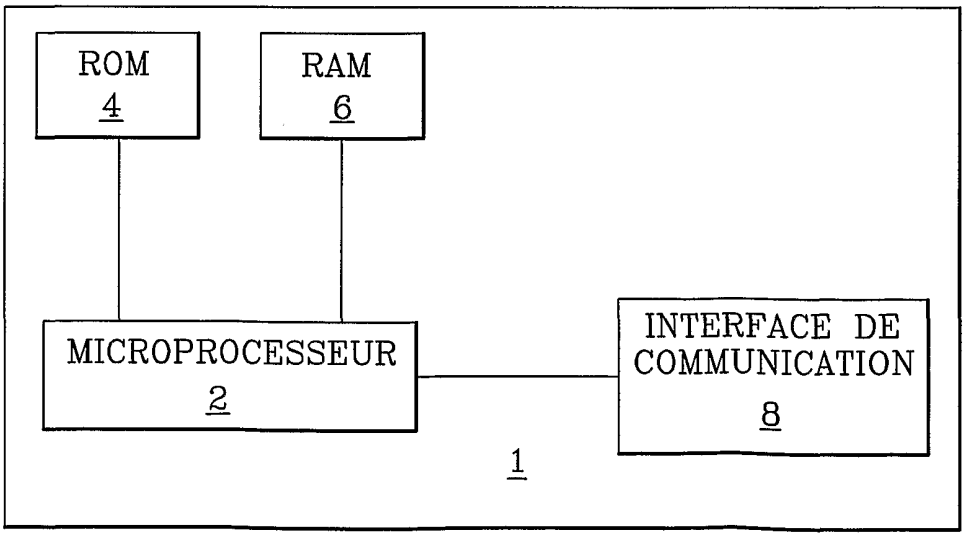


Fig. 2

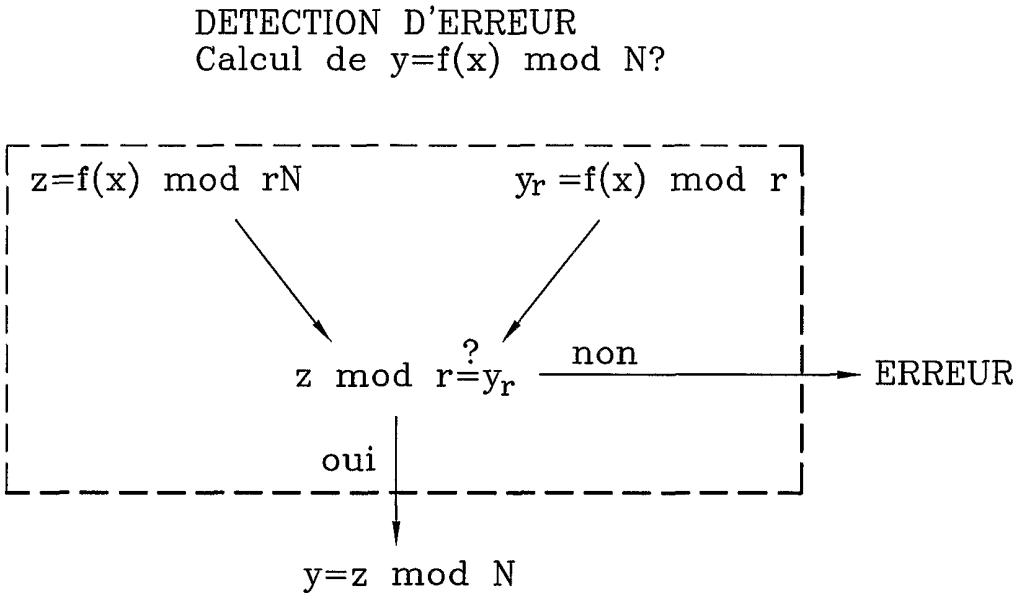

DETECTION D'ERREUR
Calcul de y=f(x) mod N?
z=f(x) mod rN
y_r =f(x) mod r
z mod r=?y_r
non
ERREUR
oui
y=z mod N

Fig. 3

INTERNATIONAL SEARCH REPORT

In tional Application No
PCT/FR 02/00113

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/30 G06F7/72

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)
IPC 7 H04L G06F

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)

WPI Data, INSPEC, PAJ, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 98 52319 A (YEDA RES & DEV ;FLEIT LOIS (US)) 19 November 1998 (1998-11-19) page 12, line 6 -page 13, paragraph 1; figure 2	1,3,6,9, 20,21, 23,26, 29,39
A	US 6 144 740 A (LAIH CHI-SUNG ET AL) 7 November 2000 (2000-11-07) column 6, line 55 -column 7, line 19	6,9,26, 29

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in 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.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 May 2002	31/05/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/FR 02/00113

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9852319	A	19-11-1998	US	5991415 A	23-11-1999
			AU	7568598 A	08-12-1998
			EP	0986873 A1	22-03-2000
			WO	9852319 A1	19-11-1998
US 6144740	A	07-11-2000	NONE		

RAPPORT DE RECHERCHE INTERNATIONALE

de internationale No
PCT/FR 02/00113

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 H04L9/30 G06F7/72

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)
CIB 7 H04L G06F

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 réalisable, termes de recherche utilisés)

WPI Data, INSPEC, PAJ, EPO-Internal

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	WO 98 52319 A (YEDA RES & DEV ;FLEIT LOIS (US)) 19 novembre 1998 (1998-11-19) page 12, ligne 6 -page 13, alinéa 1; figure 2	1,3,6,9, 20,21, 23,26, 29,39
A	US 6 144 740 A (LAIH CHI-SUNG ET AL) 7 novembre 2000 (2000-11-07) colonne 6, ligne 55 -colonne 7, ligne 19	6,9,26, 29

☐ 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'inven tion revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
"Y" document particulièrement pertinent; l'inven tion 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
"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
23 mai 2002	31/05/2002

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relati x membres de familles de brevets

D de Internationale No
PCT/FR 02/00113

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)		Date de publication
WO 9852319	A	19-11-1998	US	5991415 A	23-11-1999
			AU	7568598 A	08-12-1998
			EP	0986873 A1	22-03-2000
			WO	9852319 A1	19-11-1998
US 6144740	A	07-11-2000	AUCUN		