

(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
4 mai 2006 (04.05.2006)

PCT

(10) Numéro de publication internationale
WO 2006/045660 A2

(51) Classification internationale des brevets : **Non classée**

(21) Numéro de la demande internationale :
PCT/EP2005/054308

(22) Date de dépôt international :
1 septembre 2005 (01.09.2005)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
04/51953 1 septembre 2004 (01.09.2004) FR

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

(72) Inventeurs; et

(75) Inventeurs/Déposants (pour US seulement) : **CHEVALIER-MAMES, Benoit** [FR/FR]; La Sardanne, Les Brayes, F-13260 Cassis (FR). **CORON, Jean-Sébastien** [FR/FR]; 17, avenue d'Argenteuil, F-92600 Asnières sur Seine (FR).

(74) Mandataire : **CARDONNEL, Cédric**; Gemplus, La Vigie, Département Brevets, BP 90, 13705 La Ciotat Cedex (FR).

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

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

Publiée :

— sans rapport de recherche internationale, sera republiée dès réception de ce rapport

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.

(54) Title: ON-THE-FLY SIGNATURE GENERATION METHOD WITH SECURITY PROOF

(54) Titre : PROCEDE DE GENERATION DE SIGNATURE A LA VOLEE AVEC PREUVE DE SECURITE

(57) Abstract: The invention relates to an on-the-fly signature generation method with security proof. The invention comprises two novel electronic signature schemes with proven security. The first scheme is based on the elliptic curve discrete logarithm problem, while the second is based on any identification method without the disclosure of information. Both of the inventive schemes enable the rapid generation of an electronic signature once certain pre-calculations have been performed.

(57) Abrégé : L'invention consiste en deux nouveaux schémas de signature électronique à sécurité prouvée, le premier schéma étant basé sur le problème du logarithme discret sur courbe elliptique, le second schéma étant basé sur un procédé quelconque d'identification sans divulgation d'information. Les deux schémas de l'invention ont en commun de permettre la génération rapide d'une signature électronique une fois que certains pré-calculs ont été réalisés.



WO 2006/045660 A2

Procédé de génération de signature à la volée avec preuve
de sécurité

L'invention consiste en deux nouveaux schémas de signature électronique à sécurité prouvée, le premier
5 schéma étant basé sur le problème du logarithme discret sur courbe elliptique, le second schéma étant basé sur un procédé quelconque d'identification sans divulgation d'information. Les deux schémas de l'invention ont en commun de permettre la génération rapide d'une signature
10 électronique une fois que certains pré-calculs ont été réalisés. Cela rend l'invention particulièrement utile dans le cadre d'objets portables à faible ressource de calcul tels qu'une carte à puce.

15 Une signature électronique d'un message est un ou plusieurs nombres dépendant à la fois d'une clé secrète connue seulement de la personne signant le message, ainsi que du contenu du message à signer. Une signature électronique doit être vérifiable: il doit être possible
20 pour une tierce personne de vérifier la validité de la signature, sans que la connaissance de la clé secrète de la personne signant le message ne soit requise.

Un schéma de signature est décrit sous la forme de
25 trois éléments (GEN_S, SIGN_S, VER_S):

- GEN_S est un procédé de génération de clé publique et privée
- SIGN_S est un procédé de génération de signature
- VER_S est un procédé de vérification de
30 signature.

Il existe de nombreux procédés de signature électronique. Les plus connus sont:

- 5 - Schéma de signature RSA: c'est le schéma de signature électronique le plus largement utilisé. Sa sécurité est basée sur la difficulté de la factorisation de grands nombres.
- 10 - Schéma de signature Rabin. Sa sécurité est aussi basée sur la difficulté de la factorisation de grands nombres.
- Schéma de signature de type El-Gamal. Sa sécurité est basée sur la difficulté du problème du logarithme discret. Le problème du logarithme discret consiste à
15 déterminer, s'il existe, un entier x tel que $y=g^x$ avec y et g deux éléments d'un ensemble E possédant une structure de groupe.
- Schéma de signature Schnorr. Il s'agit d'une variante
20 du schéma de signature de type El-Gamal.

Le schéma RSA est sans doute le schéma de signature le plus connu et utilisé. Cette méthode est décrite dans
25 "A method for obtaining digital signatures and public-key cryptosystems" (Communications of the ACM, 1978) par Rivest, Shamir et Adleman.

Le procédé de génération de clé consiste à générer 2 premiers p et q , tels que $(p-1)$ et $(q-1)$ soient premiers
30 avec un exposant public e (choisi par l'utilisateur, ou initialisé à une valeur standard, par exemple $e=2^{16}+1$).

La clé secrète est alors d , tel que

$$ed=1 \text{ modulo } (p-1)*(q-1),$$

- 5 que l'on note $d=e^{(-1)} \text{ modulo } (p-1)*(q-1)$, et la clé publique est formée de $n=p*q$ et e .

Le procédé de signature, pour un message m , consiste alors, avec une fonction dite de « padding » P (le mot
10 anglais « padding » pourrait se traduire par « de remplissage », mais nous préférons, pour plus de clarté, le mot anglais dans la suite), à retourner

$$s=P(m)^d \text{ modulo } n$$

15

Le procédé de vérification d'une signature s d'un message m se résume ainsi à vérifier que $s^e \text{ modulo } n = P(m)$.

- 20 Le schéma Rabin est un schéma de signature décrit dans « Digital signatures and public-key functions as intractable as factorization », Tech. Rep. MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

- 25 C'est un procédé assez proche de RSA, qui repose sur la factorisation.

- Le procédé de génération de clé consiste à générer 2 premiers p et q . La clé secrète est alors (p,q) et la clé
30 publique est $n=p*q$.

Le procédé de signature, pour un message m , consiste alors, avec une fonction de « padding » P à retourner s , racine carrée de $P(m)$ modulo n , ce qui est facile connaissant la factorisation de n . Les algorithmes
5 permettant de faire ce calcul sont connus de l'homme de l'art.

Le procédé de vérification d'une signature s d'un message m se résume ainsi à vérifier que s^2 modulo $n =$
10 $P(m)$.

La signature Rabin est aussi sûre que la factorisation.

15 Dans l'article « How to prove yourself: Practical solutions to identification and signature problems », Proceedings of Crypto 86, Fiat et Shamir définissent la notion de procédé d'identification sans divulgation d'information (en anglais « zero-knowledge protocol »),
20 ainsi qu'une heuristique, dite heuristique de Fiat-Shamir, qui permet de transformer un procédé d'identification sans divulgation d'information en schéma de signature.

25 Un procédé d'identification sans divulgation d'information comprend :

- un procédé de génération de clé publique et privée
- un protocole d'identification, qui peut
30 généralement se décrire à l'aide de trois étapes, en un prouveur P et un vérifieur V ; pour cela,

on définit 3 fonctions publiquement connus A, B et C ; le protocole est alors:

- 5 - étape 1 : P génère un nombre aléatoire r ,
 calcule $x=A(r)$ et envoie x à V
- étape 2 : V génère un nombre aléatoire g et
 l'envoie à P
- 10 - étape 3 : P calcule, à l'aide de son secret s ,
 le nombre $y=B(s,g,r)$, et l'envoie à V. V peut
 alors vérifier si $x=C(y,g)$, auquel cas
 l'identification est acceptée.

15 Le protocole d'identification peut également
 comporter la vérification que les éléments sont
 dans des ensembles connus. Un moyen simple de
 vérifier l'appartenance de ces éléments à ces
 ensembles est de ne définir les fonctions A, B, C
 que sur ces ensembles. Un exemple est donné ci-
 dessous par le schéma « Poupard Stern »

20

 L'heuristique Fiat-Shamir consiste à supprimer
 l'interactivité du procédé d'identification sans
 divulcation d'information: pour cela, on utilise une
 fonction de hachage (« hash », en anglais), appelée G. Le
25 signataire est appelé S, tandis que le vérifieur est
 appelé V.

 Le schéma de signature alors obtenu est alors
 caractérisé par un procédé de génération de clé qui est
30 le procédé de génération de clé du procédé

d'identification sans divulgation d'information, et par un procédé de génération de signature qui est:

- 5 - générer un nombre aléatoire r et calculer $x=A(r)$;
- calculer $g=G(m,x)$, où m est le message à signer ;
- calculer $y=B(s,g,r)$;
- la signature alors obtenue est (x,y) .

10

Enfin, le procédé de vérification de la signature (x,y) du message m consiste à calculer $g=G(m,x)$, puis à vérifier que $x=C(y,g)$.

- 15 On dit qu'un schéma de signature est de type « à la volée » (« on the fly », en anglais) lorsque la génération de la signature peut se décomposer en deux phases distinctes : une première phase dite de pré-calcul, durant laquelle une valeur indépendante du
- 20 message à signer est calculée, et la phase de génération de signature, durant laquelle une signature d'un message m est calculée en utilisant la valeur pré-calculée, cette dernière phase devant pouvoir s'exécuter rapidement. Les schémas de signature « à la volée » sont donc
- 25 particulièrement utiles dans le cadre d'objets portables à faible ressource de calcul tels qu'une carte à puce. De tels schémas permettent la génération rapide de la signature par l'objet portable, alors que cela ne serait pas possible pour un schéma de signature classique
- 30 demandant plus de ressources de calculs.

On voit ainsi aisément que les schémas basés sur l'heuristique Fiat-Shamir sont des schémas de signatures « à la volée » dès lors que l'évaluation de la fonction G (la fonction de hachage) et de la fonction B (pour le calcul de y) est rapide, ce qui est le cas pour les 5 procédé d'identification sans divulgation d'information classiques (Fiat-Shamir, Girault-Poupard-Stern, Poupard-Stern). En effet, l'étape consistant à générer un nombre aléatoire r et à calculer $x=A(r)$ est indépendante du 10 message, et peut donc se faire sans la connaissance de celui ci, lors d'un pré-calcul.

Girault-Poupard-Stern est décrit dans "An identity-based identification scheme based on discrete logarithms 15 modulo a composite number", Proceedings of EUROCRYPT 90.

Fiat-Shamir est décrit dans « How to prove yourself: Practical solutions to identification and signature problems", Proceedings of Crypto 86.

20

Poupard-Stern est détaillé ci-après.

Malheureusement, RSA et Rabin ne disposent pas de cette propriété très intéressante de signature « à la 25 volée ». Une variante de RSA et de Rabin proposant cet avantage est l'œuvre de cette invention.

Poupard-Stern est un procédé d'identification sans divulgation d'information, publiée dans « On the fly 30 signatures based on factoring », dans ACM Conference on Computer and Communications Security 1999, par Poupard et

Stern. Poupard-Stern peut être transformé en procédé de signature, grâce à l'heuristique Fiat-Shamir.

Le procédé de génération de clé consiste à générer 2
5 premiers p et q . La clé publique comprend alors un module $n=p*q$, dont la factorisation est gardée secrète, et une base a . La clé secrète est $s=n-(p-1)(q-1)$.

Le protocole d'identification est alors:

- 10
- Le prouveur P génère un nombre aléatoire r
 - P calcule $x=a^r$ modulo n et l'envoie au vérifieur V
 - V génère un nombre aléatoire g et l'envoie à P .
 - P calcule $y=r+s*g$ et l'envoie à V

15

Le vérifieur V peut alors vérifier que $a^y=x*a^{(n*g)}$ modulo n .

De plus, il est vérifié que y est dans le bon
20 intervalle de valeur. Plus précisément, le nombre r étant généré entre 0 et R , le nombre g étant de taille $||g||$ bits, et le nombre s étant de taille $||s||$ bits, il est vérifié que : $0 \leq y \leq R+2^{(||s||+||k||)}$. Cet intervalle, ainsi que les tailles des paramètres, sont décrits plus
25 précisément dans l'article de Poupard et Stern.

On dit qu'un schéma de signature est à sécurité prouvée si on peut, par une preuve mathématique, utiliser un éventuel attaquant contre ce schéma de signature (plus
30 précisément, les signatures forgées, c'est à dire contrefaites par cet attaquant) pour résoudre un problème

difficile, comme le logarithme discret ou la factorisation.

Les preuves peuvent être non strictement réductibles
5 (« loose » en anglais) ou strictement réductibles
(« tight » en anglais). Les termes anglais seront
préférés dans la suite.

Une preuve « loose » utilise un attaquant et résout
10 le problème difficile avec une probabilité faible
comparée à celle de l'attaquant. Au contraire, une preuve
« tight » résout le problème avec une probabilité très
proche de celle de l'attaquant. Ainsi, une preuve
« tight » est un bien meilleur gage de sécurité.

15

Les schémas de signature basés sur l'heuristique
Fiat-Shamir possèdent une preuve de sécurité. Cette
preuve est réalisée grâce à l'article « Security proofs
for signature schemes », Proceedings of EUROCRYPT 96, où
20 Pointcheval et Stern introduisent le lemme de bifurcation
("forking lemma", en anglais), qui permet de prouver la
sécurité de schémas basés sur l'heuristique Fiat-Shamir,
dans le modèle de l'oracle aléatoire.

25 Malheureusement, les preuves dues au « forking
lemma » n'assurent qu'une sécurité « loose ». Il n'existe
pas de moyen connu de prouver un schéma basé sur
l'heuristique Fiat-Shamir avec une preuve « tight ».
Ainsi, la preuve de sécurité de Poupard-Stern, reposant
30 sur le lemme de bifurcation, n'est que « loose ».

Le modèle de l'oracle aléatoire est un modèle idéal dans lequel toute fonction de hachage est considérée comme parfaitement aléatoire. Comme dans la pratique, une fonction de hachage n'est pas un objet parfaitement aléatoire, on considère généralement qu'une preuve dans le modèle de l'oracle aléatoire est une indication que le schéma est correctement construit, mais que cela ne donne pas une garantie parfaite de la sécurité du schéma lors de son utilisation pratique.

10

Au contraire, on dit qu'un schéma cryptographique possède une preuve de sécurité dans le modèle standard lorsque sa sécurité peut être prouvée sans faire l'hypothèse du caractère parfaitement aléatoire des fonctions de hachage. Une telle preuve de sécurité est particulièrement utile car elle permet d'avoir une confiance parfaite dans la sécurité du schéma lors de son utilisation pratique.

20 Dans la publication "Improved online/offline signature schemes" par Shamir et Tauman (Proceedings of Crypto 01), les auteurs décrivent un moyen de conversion générique permettant d'obtenir un schéma de signature « à la volée » à partir d'un schéma de signature quelconque.

25 L'avantage de cette conversion est qu'elle préserve la sécurité du schéma de signature : si le schéma initial possède une preuve de sécurité dans le modèle standard, alors le schéma de signature « à la volée » obtenue possède aussi une preuve de sécurité dans le modèle

30 standard. L'inconvénient de cette méthode est qu'elle double la taille de la clef publique ainsi que la taille

de la signature, et qu'elle augmente aussi le temps de vérification de la signature. Le temps total de génération de la signature (pré-calcul+génération) est lui-même augmenté.

5

Le premier schéma de l'invention consiste en un premier schéma de signature à la volée avec une preuve « tight » de sécurité dans le modèle standard et ne présentant pas les inconvénients cités précédemment. En effet, dans ce premier procédé de l'invention, la taille
10 de la clef publique et de la signature ne sont pas augmentées, et le temps de vérification de la signature n'est pas modifié.

15 Le premier procédé de l'invention est basé sur le schéma de signature inventé par Boneh and Boyen ("Short signatures without random oracles", Proceedings of Eurocrypt 2004). Ce schéma de signature est basé sur le calcul du couplage dans une courbe elliptique, et possède
20 une preuve de sécurité « tight » dans le modèle standard. En revanche, ce schéma ne possède pas la propriété d'être « à la volée », c'est à dire que la génération de la signature est coûteuse en temps de calcul et ne peut pas se décomposer en deux phases telles que définies
25 précédemment.

Le calcul du couplage de deux points définis sur une courbe elliptique est un procédé permettant la mise en oeuvre de protocoles possédant des propriétés
30 intéressantes sur le plan cryptographique. A titre d'exemple, on peut citer un protocole de chiffrement basé

sur l'identité (Boneh et Franklin, « Identity-based encryption from the Weil pairing », Proceedings of Crypto 2001) ainsi qu'un protocole d'échange de clef entre trois parties (A. Joux, "A one-round protocol for tripartite Diffie-Hellman," Algorithm Number Theory Symposium -- ANTS-IV, lecture Notes on Computer Science, volume 1838, Springer-Verlag (2000), pp. 385-394).

Une courbe elliptique est un ensemble de points ayant des caractéristiques communes. Etant donné deux points A et B appartenant à une courbe elliptique, il est possible d'additionner ces deux points pour en obtenir un troisième $C=A+B$. De plus, étant donné un point A et un entier k, il est possible de calculer $k.P=P+P+...+P$ (l'opération étant répétée k fois). Cette opération s'appelle multiplication scalaire par un entier k. Etant donné deux points A et B appartenant à une courbe elliptique, il est possible de calculer le couplage entre les points A et B, noté $\langle A, B \rangle$. Pour plus de détails sur les courbes elliptiques, on pourra se reporter notamment à la publication FR 2 828 779. La mise en oeuvre classique d'une opération de couplage de type $C = \langle A, B \rangle$ est détaillée notamment dans le document "Identity based encryption from the Weil pairing" D. Boneh et M. Franklin, Proceedings of Crypto 2001, disponible à l'adresse :<http://crypto.stanford.edu/~dabo/abstracts/ibe.html>). Les points A, B que l'on cherche à coupler sont par exemple une clé, publique ou privée, et un message, à chiffrer ou à déchiffrer, dans le cadre d'un protocole de cryptographie à clés asymétriques.

Le schéma de signature inventé par Boneh et Boyen sur lequel est basé le premier procédé de l'invention, est défini de la façon suivante :

5 G_1 et G_2 sont deux générateurs de courbes elliptiques E_1 et E_2 , respectivement. Une réalisation possible du schéma consiste à prendre $E_1=E_2$ et $G_1=G_2$. On suppose que ces deux générateurs sont d'ordre p , où p est un nombre premier. Les courbes elliptiques E_1 et E_2 , les
10 points G_1 et G_2 , ainsi que p , sont des paramètres du système. x et y sont deux entiers définis modulo p , et U est tel que $U=x.G_2$ et V tel que $V=y.G_2$. La clef publique est (U,V) et la clef privée est (x,y) .

15 Pour générer une signature, on génère un aléa r compris entre 0 et $p-1$, et la signature est alors (s,r) , tel que :

$$s=1/(x+H(m)+y*r).G_1$$

20

dans laquelle $H(m)$ est le haché du message m .

La vérification de la validité d'une signature s'effectue de la manière suivante : on vérifie l'égalité suivante :

25 $\langle s, U+H(m).G_2+r.V \rangle = \langle G_1, G_2 \rangle$

On voit d'après la description précédente du schéma de signature de Boneh et Boyen que ce schéma n'est pas du type « à la volée ». En effet, pour calculer la signature
30 (s,r) , il est nécessaire de connaître le message m , et la

multiplication scalaire ainsi définie (ie,
 $s=1/(x+H(m)+y*r).G1$) ne peut pas être pré-calculée.

Le premier schéma de l'invention consiste à modifier
5 le schéma de signature précédent pour le rendre du type
« à la volée ». Le procédé de génération de la clef
publique et de la clef privée n'est pas modifié par
rapport au schéma original. Le procédé de vérification de
la signature n'est pas modifié non plus. Seul le procédé
10 de génération de signature est modifié de la façon
suivante selon deux phases.

Dans une première phase, dite phase de pré-calcul,
on génère un aléa r' compris entre 0 et $p-1$, et on
15 calcule le point :

$$s=1/(x+y*r').G1$$

Dans une deuxième phase, étant donné le message m ,
20 on calcule l'entier:

$$r=r'-H(m)/y \text{ modulo } p$$

La signature est alors égale à (s,r) .
25

Plus précisément, ce premier schéma de signature à
sécurité prouvé « tight » et « à la volée » destiné à un
composant électronique est tel qu'il utilise les procédés
suivants :

- GEN_NOU est le procédé de génération de la clé publique et privée selon le schéma de signature de Boneh et Boyen,
- SIGN_NOU est le procédé de génération de signature d'un message m à signer et consiste à :
 - 1) générer un aléa r' compris entre 0 et $p-1$, l'entier p étant un paramètre du schéma ;
 - 2) pré-calculer le point $s=1/(x+y*r')$.G1, le point G1 étant un générateur de la courbe elliptique E1, le point G1 et la courbe elliptique G1 étant deux paramètres du schéma, les entiers x et y formant la clef privée du schéma ;
 - 3) calculer, une fois le message m connu, l'entier $r=r'-H(m)/y$ modulo p ;
 - 4) renvoyer la signature (s,r) ;
- VER_NOU est le procédé de vérification selon le schéma de signature de Boneh et Boyen.

20

On voit que dans le schéma décrit précédemment, le point s est indépendant du message m et peut donc être pré-calculé. Ensuite, lors de la deuxième phase, lorsque le signataire reçoit le message m , il peut calculer l'entier $r=r'-H(m)/y$ modulo p . Comme cette opération ne requiert qu'une division et une soustraction, elle peut s'effectuer très rapidement. Le premier procédé de l'invention est donc du type « à la volée »

25

De plus, comme la distribution de probabilité des signatures générées n'a pas été modifiée, le premier

30

procédé de l'invention définit un schéma de signature possédant le même niveau de sécurité que le schéma original, c'est à dire qu'il possède une preuve de sécurité « tight » dans le modèle standard.

5

Le premier procédé de l'invention comprend une première variante destinée à réduire le temps de génération de la signature. Dans cette première variante, on suppose que le signataire possède en mémoire l'entier $y' = -1/y$ modulo p . Dans ce cas, la génération de la signature s'effectue en calculant $r = r' + H(m) * y'$ modulo p . L'avantage de la première variante est qu'elle remplace une division modulaire par une multiplication modulaire dont le calcul est généralement plus rapide. Comme la distribution de probabilité des signatures générées n'a pas été modifiée, cette variante possède exactement le même niveau de sécurité.

Le premier procédé de l'invention comprend une deuxième variante destinée à réduire encore le temps de génération de la signature. Dans cette deuxième variante, le signataire génère un aléa r' de grande taille (par exemple de taille 400 bits si la taille de p est de 160 bits). Dans cette deuxième variante, le signataire calcule $r = r' + H(m) * y$, le calcul s'effectuant sur les entiers et non modulo p . Comme le calcul d'une addition et d'une multiplication sur les entiers est plus rapide qu'une addition et une multiplication modulo p , la deuxième variante s'exécute plus rapidement. De plus, on peut montrer que le niveau de sécurité du schéma n'est

pas modifié de façon significative si la taille de r' est suffisamment grande.

Le deuxième schéma de l'invention consiste en un
5 schéma de signature basé sur l'heuristique Fiat-Shamir, mais avec le nouvel avantage que la preuve est « tight ». Le schéma gardera toutefois la bonne propriété d'être « à la volée ».

10 L'invention utilise ainsi n'importe quel procédé d'identification sans divulgation d'information. Elle utilise aussi n'importe quel schéma de signature S , pour lequel il soit impossible pour un attaquant, étant donné k messages, de procéder à une forge sur l'un quelconque
15 de ces messages. Enfin, l'invention utilise deux fonctions de hachage G et H .

Plus précisément, le deuxième schéma de l'invention est un schéma de signature à sécurité prouvée « tight » et
20 « à la volée » destiné à un composant électronique, utilisant deux fonctions de hachage G et H , un procédé d'identification sans divulgation d'information ZK , donné sous la forme (GEN_ZK, A, B, C) et un schéma de signature S , donné sous la forme $(GEN_S, SIGN_S, VER_S)$, où

- 25
- GEN_ZK est le procédé de génération de clé publique et privée de ZK ;
 - A, B, C sont les fonctions définissant le protocole d'identification de ZK ;
 - GEN_S est le procédé de génération de clé
30 publique et privée de S ;

- SIGN_S est le procédé de génération de signature de S ;
 - VER_S est le procédé de vérification de signature de S ;
- 5 ledit schéma de signature de l'invention est tel qu'il utilise les procédés suivants :
- GEN_INV est le procédé de génération de clé publique et privée, qui consiste à exécuter GEN_ZK et GEN_S
 - 10 • SIGN_INV est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r et calculer $x=A(r)$;
 - o pré-calculer $h=H(x)$;
 - 15 o pré-calculer la signature de h : $z=SIGN_S(h)$;
 - o calculer $g=G(m,z)$, où m est le message à signer, une fois celui-ci connu;
 - o calculer $y=B(s,g,r)$,
 - 20 et ainsi à obtenir la signature (z,y)
 - VER_INV est le procédé de vérification de signature (z,y) du message m, et consiste à :
 - calculer $g=G(m,z)$;
 - 25 ▪ calculer $x=C(y,g)$;
 - calculer $h=H(x)$;
 - vérifier que z est une signature valide de h à l'aide de VER_S.
- 30 Comme on peut le voir aisément, ce schéma est « à la volée », car les étapes premières de la signature

(précisément, jusqu'au calcul de g non inclus) peuvent être faites sans connaître le message. Les étapes finales étant très rapides, le schéma est « à la volée ». Plus précisément, ces étapes peuvent être réalisées avant la
5 connaissance du message m , car elles sont indépendantes de celui ci : par un tel procédé, le temps de la signature une fois le message reçu est grandement diminué.

10 On peut alors prouver que ce schéma possède une preuve de sécurité « tight », permettant de résoudre, en utilisant un attaquant du schéma, soit le problème sur lequel repose le procédé d'identification sans divulgation d'information, soit le problème sur lequel
15 repose le schéma de signature S .

Un exemple non limitatif est celui où S est le RSA et où le procédé d'identification sans divulgation d'information est Poupard-Stern.

20

Plus précisément, une telle combinaison nous permet d'obtenir un schéma de signature utilisant le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation d'information, et le schéma RSA, comme
25 étant le schéma de signature, schéma de l'invention tel qu'il utilise les procédés suivants :

- GEN_INV est le procédé de génération de clé publique et clé secrète et consiste à :
 - générer un paramètre a appelé base,

- générer deux premiers p et q , tel que $(p-1)$ et $(q-1)$ soient premiers avec un exposant public e , et
 - retourner $(n=p*q, e)$ comme clé publique, et $(d=e^{(-1)} \text{ modulo } (p-1)*(q-1), s=n-(p-1)*(q-1))$ comme clé privée, la taille de s en bit étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- SIGN_INV est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x=a^r \text{ modulo } n$;
 - o pré-calculer $h=H(x)$;
 - o pré-calculer la signature de h : $z=h^d \text{ modulo } n$;
 - o calculer $g=G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - o calculer $y=r+s*g$ et ainsi obtenir la signature (z, y) ;
- VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste à :
 - calculer $g=G(m, z)$;
 - calculer $x=a^y * a^{(-n*g)} \text{ modulo } n$;
 - calculer $h=H(x)$;

- vérifier que $z^e = h$ modulo n et que y est dans le bon intervalle $[0, R+2^{(|s|+|g|)}]$.

5 Comme on peut le voir aisément, ce schéma est « à la volée », car les étapes premières de la signature (précisément, jusqu'au calcul de g non inclus) peuvent être faites sans connaître le message, et donc avant la réception de celui ci, comme une étape de pré calcul. Les
10 étapes finales étant très rapides, le schéma est « à la volée ».

 La sécurité de ce schéma repose sur la difficulté de trouver une racine e -ieme modulo n . La preuve est
15 « tight ».

Ce schéma permet donc de faire une variante du RSA, mais avec un caractère « à la volée ».

 L'avantage en terme de vitesse est grand : le calcul
20 en ligne (c'est à dire une fois reçu le message m) n'est qu'un calcul de haché et une multiplication. Au contraire, le RSA nécessite le calcul d'une exponentiation d'au moins 1024 bits. Ainsi, notre système est environ 1024 fois plus rapide que le RSA classique,
25 tout en gardant la même sécurité « tight ».

 Enfin, on remarquera qu'il est possible, comme dans le RSA, de définir d non pas modulo $(p-1)(q-1)$, mais modulo $\text{PPCM}(p-1, q-1)$, où PPCM est le plus petit commun
30 multiple.

Alors, $ed=1$ modulo $\text{PPCM}(p-1, q-1)$. Le reste est inchangé.

Un autre exemple est celui où S est la signature de Rabin et où le procédé d'identification sans divulgation
5 d'information est Poupard-Stern.

Plus précisément, une telle combinaison nous permet d'obtenir un schéma de signature qui utilise le schéma Poupard-Stern, comme étant le procédé d'identification
10 sans divulgation d'information, et le schéma Rabin, comme étant le schéma de signature, tel que le schéma de signature utilise les procédés suivants :

- GEN_INV est le procédé de génération de clé publique et clé secrète et consiste à :
 - 15 ▪ générer un paramètre a appelé base ;
 - générer deux premiers p et q ;
 - retourner $n=p*q$ comme clé publique, et $(p, q, s=n-(p-1)*(q-1))$ comme clé privée, la taille de s en bit étant connue, et
20 notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- 25 • SIGN_INV est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x=a^r$ modulo n
 - o pré-calculer $h=H(x)$

- o pré-calculer, à l'aide de p et q , la signature z comme étant une racine carrée de h modulo n
- o calculer $g=G(m,z)$, où m est le message à signer, une fois celui-ci connu ;
- o calculer $y=r+s*g$ et ainsi à obtenir la signature (z,y)
- VER_INV est le procédé de vérification de signature (z,y) du message m , et consiste à :
 - calculer $g=G(m,z)$;
 - calculer $x=a^y * a^{(-n*g)} \text{ modulo } n$;
 - calculer $h=H(x)$;
 - vérifier que $z^2 = h \text{ modulo } n$ et que y est dans le bon intervalle $[0, R+2^{(||s||+||g||)}]$.

Comme on peut le voir aisément, ce schéma est « à la volée », car les étapes premières de la signature (précisément, jusqu'au calcul de g non inclus) peuvent être faites sans connaître le message, et donc avant la réception de celui ci, comme une étape de pré calcul. Les étapes finales étant très rapides, le schéma est « à la volée ».

25

La sécurité de ce schéma repose sur la difficulté de factoriser un grand nombre. La preuve est « tight ».

Ce schéma permet donc de faire une variante du RSA, mais avec un caractère « à la volée ».

30

De même que pour l'exemple précédent, l'avantage en terme de vitesse est grand : le calcul en ligne n'est qu'un calcul de haché et une multiplication. Au contraire, le Rabin nécessite le long calcul d'une racine carrée. Ainsi, notre système est environ 1024 fois plus rapide que le Rabin classique, tout en gardant la même sécurité « tight ».

Certaines variantes habituelles peuvent être appliquées à notre deuxième partie de l'invention. Notamment, dans les procédés de vérification de nos deux exemples, la quantité $\alpha = a^{(-n)} \text{ modulo } n$ peut être précalculée, rendant ainsi le calcul de x plus rapide.

Cela nous donne les deux variantes suivantes :

La première variante consiste en un schéma de signature qui utilise le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation d'information, et le schéma RSA comme étant le schéma de signature, tel que ce schéma selon la première variante utilise les procédés suivants

- GEN_INV est le procédé de génération de clé publique et privée et consiste à :
 - générer un paramètre a appelé base,
 - générer deux premiers p et q , tel que $(p-1)$ et $(q-1)$ soient premiers avec un exposant public e , et
 - retourner $(n=p*q, e, \alpha=a^{(-n)} \text{ modulo } n)$ comme clé publique, et $(d=e^{(-1)} \text{ modulo } (p-1)*(q-1), s=n-(p-1)*(q-1))$

- comme clé privée, la taille de s en bit étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant une valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- 5
- $SIGN_INV$ est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x = a^r \text{ modulo } n$;
 - 10 o pré-calculer $h = H(x)$;
 - o pré-calculer la signature de h : $z = h^d \text{ modulo } n$;
 - o calculer $g = G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - 15 o calculer $y = r + s * g$,
et ainsi à obtenir la signature (z, y) ;
 - VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste à :
 - 20 o calculer $g = G(m, z)$;
 - o calculer $x = a^y * \alpha^g \text{ modulo } n$;
 - o calculer $h = H(x)$;
 - o vérifier que $z^e = h \text{ modulo } n$ et que y est dans le bon intervalle $[0, R + 2^{(||s|| + ||g||)}]$.
 - 25

La seconde variante consiste en un schéma de signature qui utilise le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation d'information, et le schéma Rabin, comme étant le schéma

30

de signature, tel que ce schéma selon la seconde variante utilise les procédés suivants :

- GEN_INV est le procédé de génération de clé publique et clé secrète et consiste à :
 - 5 ▪ générer un paramètre a appelé base ;
 - générer deux premiers p et q ;
 - retourner $(n=p*q, \alpha=a^{(-n)} \text{ modulo } n)$ comme clé publique, et $(p, q, s=n-(p-1)*(q-1))$ comme clé privée, la taille de s en bit étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- SIGN_INV est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x=a^r \text{ modulo } n$
 - 20 o pré-calculer $h=H(x)$
 - o pré-calculer, à l'aide de p et q , la signature z comme étant une racine carrée de $h \text{ modulo } n$
 - o calculer $g=G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - 25 o calculer $y=r+s*g$
 - et ainsi à obtenir la signature (z, y)
- VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste :
 - 30 ▪ calculer $g=G(m, z)$;
 - calculer $x=a^y * \alpha^g \text{ modulo } n$;

- calculer $h=H(x)$, et enfin
- vérifier que $z^2 = h$ modulo n et que y est dans le bon intervalle $[0, R+2^{(|s|+|g|)}]$.

5

De plus, la clé peut être compressée, c'est à dire stockée sous une forme prenant moins de place. Ainsi, on voit aisément dans l'exemple avec Rabin que s peut ne pas être stocké, mais directement recalculé à chaque utilisation par une simple soustraction.

De manière préférentielle, dans l'exemple avec le RSA, la clé privée est générée sous la forme :

$(d=e^{-1} \text{ modulo } (p-1)*(q-1), k=(e*d-1)/((p-1)*(q-1)))$,
 et dans lequel, au moment de la signature, s est recalculé par la formule $s=n-(ed-1)/k$.

k étant petit (car $0 \leq k \leq e$ et e petit), le stockage sous cette forme de la clé privée est d'autant moins coûteux en mémoire.

D'une autre manière préférentielle, dans l'exemple avec le Rabin, la clé privée est générée sous la forme :

(p, q)
 et dans lequel, au moment de la signature, s est recalculé par la formule $s=p*q-(p-1)*(q-1)$.

Il est à noter que dans le RSA, de manière préférentielle, d est défini comme : $d=e^{-1} \text{ modulo } \text{PPCM}(p-1, q-1)$.

Dans le RSA, d'une autre manière préférentielle, d est défini comme :

$d = e^{-1} \text{ modulo } u * \text{PPCM}(p-1, q-1)$, où u est un nombre entier positif non nul.

5

Etant parfaitement généralistes, toutes les variantes de cette deuxième partie de l'invention ne peuvent être décrites en détail. On citera notamment l'utilisation possible des signatures RSA, Rabin,
10 Paillier, El Gamal, Goh-Jarecki comme signature S ; parallèlement, on citera l'utilisation possible des procédés d'identification sans divulgation d'information Fiat-Shamir, Poupard-Stern, Girault-Poupard-Stern.

15 La signature de Paillier est décrite « Public-key cryptosystems based on composite degree residuosity classes », Proceedings of EUROCRYPT 99.

La signature El Gamal est décrite « A public key
20 cryptosystem and a signature scheme based on discrete logarithms », IEEE Transactions on Information Theory, 1985.

De plus, la signature de Goh-Jarecki est décrite
25 dans « A signature scheme as secure as the Diffie-Hellman problem », Proceedings of EUROCRYPT 03.

En outre, les deux schémas à sécurité prouvé « tight » et « à la volée » destiné à un composant
30 électronique de cette invention sont mis en oeuvre un composant électronique portable du type carte à puce.

Enfin, le calcul de ces deux schémas s'effectue sur un module TPM pour « Trusted Platform Module » installé dans un PC non sécurisé.

REVENDICATIONS

1. Schéma de signature à sécurité prouvé « tight » et « à la volée » destiné à un composant électronique caractérisé en ce que le schéma de signature utilise les
5 procédés suivants :

- GEN_NOU est le procédé de génération de la clé publique et privée selon le schéma de signature de Boneh et Boyen,
- SIGN_NOU est le procédé de génération de
10 signature d'un message m à signer et consiste à :
 - 1) générer un aléa r' compris entre 0 et $p-1$, l'entier p étant un paramètre du schéma ;
 - 2) pré-calculer le point $s=1/(x+y*r').G1$, le
15 point $G1$ étant un générateur de la courbe elliptique $E1$, le point $G1$ et la courbe elliptique $G1$ étant deux paramètres du schéma, les entiers x et y formant la clef privée du schéma ;
 - 3) calculer, une fois le message m connu,
20 l'entier $r=r'-H(m)/y$ modulo p ;
 - 4) renvoyer la signature (s,r) ;
- VER_NOU est le procédé de vérification selon le schéma de signature de Boneh et Boyen.

25

2. Schéma de signature selon la revendication 1, caractérisé en ce que le signataire possède en mémoire l'entier $y'=-1/y$ modulo p , dans lequel l'étape 3) de la génération de la signature selon la revendication 1 est
30 remplacée par l'étape suivante :

3') calculer une fois le message m connu l'entier $r=r'+H(m)*y$ modulo p .

3. Schéma de signature selon la revendication 1,
5 caractérisé en ce que l'étape 1) de la génération de la signature est remplacée par l'étape :

10 1'') générer un aléa compris entre 0 et B , l'entier B étant un paramètre du système choisi très supérieur à p ,

et tel que l'étape 3) de la génération de la signature est remplacée par l'étape suivante :

15 3'') calcule une fois le message m connu l'entier $r=r'+H(m)*y$

4. Schéma de signature à sécurité prouvé « tight » et « à la volée » destiné à un composant électronique,
20 utilisant deux fonctions de hachage G et H , un procédé d'identification sans divulgation d'information ZK , donné sous la forme (GEN_ZK, A, B, C) et un schéma de signature S , donné sous la forme $(GEN_S, SIGN_S, VER_S)$, où

- 25 • GEN_ZK est le procédé de génération de clé publique et privée de ZK ,
- A, B, C sont les fonctions définissant le protocole d'identification de ZK
- GEN_S est le procédé de génération de clé publique et privée de S
- 30 • $SIGN_S$ est le procédé de génération de signature de S

- VER_S est le procédé de vérification de signature de S
caractérisé en ce que le schéma de signature utilise les procédés suivants :
- 5 • GEN_INV est le procédé de génération de clé publique et privée, qui consiste à exécuter GEN_ZK et GEN_S
- SIGN_INV est le procédé de génération de signature, et consiste à :
 - 10 o générer un nombre aléatoire r et calculer $x=A(r)$;
 - o pré-calculer $h=H(x)$;
 - o pré-calculer la signature de h : $z=SIGN_S(h)$;
 - 15 o calculer $g=G(m,z)$, où m est le message à signer, une fois celui-ci connu;
 - o calculer $y=B(s,g,r)$,
 et ainsi à obtenir la signature (z,y)
- VER_INV est le procédé de vérification de signature (z,y) du message m, et consiste à :
 - 20 ▪ calculer $g=G(m,z)$;
 - calculer $x=C(y,g)$;
 - calculer $h=H(x)$;
 - vérifier que z est une signature valide
25 de h à l'aide de VER_S.

5. Schéma de signature, selon la revendication 4, caractérisé en ce qu'il utilise le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation
30 d'information, et le schéma RSA, comme étant le schéma de signature tel qu'il utilise les procédés suivants :

- GEN_INV est le procédé de génération de clé publique et clé secrète et consiste à :

 - générer un paramètre a appelé base,
 - générer deux premiers p et q , tel que
5 $(p-1)$ et $(q-1)$ soient premiers avec un exposant public e , et
 - retourner $(n=p*q, e)$ comme clé publique, et $(d=e^{-1} \text{ modulo } (p-1)*(q-1), s=n-(p-1)*(q-1))$ comme clé privée, la taille
10 de s en bit étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- SIGN_INV est le procédé de génération de signature, et consiste à :

 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x=a^r \text{ modulo } n$;
 - o pré-calculer $h=H(x)$;
 - o pré-calculer la signature de h : $z=h^d \text{ modulo } n$;
20
 - o calculer $g=G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - o calculer $y=r+s*g$
 - et ainsi obtenir la signature (z, y) ;
25
- VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste à :

 - calculer $g=G(m, z)$;
 - calculer $x=a^y * a^{(-n*g)} \text{ modulo } n$;
 - calculer $h=H(x)$;
30

- vérifier que $z^e = h$ modulo n et que y est dans le bon intervalle $[0, R+2^{(||s||+||g||)}]$.

5 6. Schéma de signature selon la revendication 4, caractérisé en ce qu'il utilise le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation d'information, et le schéma RSA comme étant le schéma de signature, tel qu'il utilise les procédés suivants

- 10 • GEN_INV est le procédé de génération de clé publique et privée et consiste à :
 - générer un paramètre a appelé base,
 - générer deux premiers p et q , tel que $(p-1)$ et $(q-1)$ soient premiers avec un
 - 15 exposant public e , et
 - retourner $(n=p*q, e, \alpha=a^{(-n)} \text{ modulo } n)$ comme clé publique, et $(d=e^{(-1)} \text{ modulo } (p-1)*(q-1), s=n-(p-1)*(q-1))$ comme clé privée, la taille de s en bit
 - 20 étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant une valeur maximale du nombre aléatoire dans le procédé de génération de signature ;
- 25 • SIGN_INV est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x=a^r \text{ modulo } n$;
 - o pré-calculer $h=H(x)$;
 - 30 o pré-calculer la signature de h : $z=h^d \text{ modulo } n$;

- o calculer $g=G(m,z)$, où m est le message à signer, une fois celui-ci connu;
- o calculer $y=r+s*g$,
et ainsi à obtenir la signature (z,y) ;
- 5 • VER_INV est le procédé de vérification de signature (z,y) du message m , et consiste à :
 - calculer $g=G(m,z)$;
 - calculer $x=a^y * \alpha^g$ modulo n ;
 - calculer $h=H(x)$;
 - 10 ▪ vérifier que $z^e = h$ modulo n et que y est dans le bon intervalle $[0, R+2^{(||s||+||g||)}]$.

7. Schéma de signature selon la revendication 4,
15 caractérisé en ce qu'il utilise le schéma Poupard-Stern, comme étant le procédé d'identification sans divulgation d'information, et le schéma Rabin, comme étant le schéma de signature, tel qu'il utilise les procédés suivants :

- GEN_INV est le procédé de génération de clé
20 publique et clé secrète et consiste à :
 - générer un paramètre a appelé base ;
 - générer deux premiers p et q ;
 - retourner $n=p*q$ comme clé publique, et
($p,q,s=n-(p-1)*(q-1)$) comme clé privée,
25 la taille de s en bit étant connue, et notée $||s||$, ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération
 - 30 de signature ;
- SIGN_INV est le procédé de génération de signature, et consiste à :

- o générer un nombre aléatoire r , entre 0 et R , et calculer $x = a^r \text{ modulo } n$
 - o pré-calculer $h = H(x)$
 - o pré-calculer, à l'aide de p et q , la signature z comme étant une racine carrée de h modulo n
 - o calculer $g = G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - o calculer $y = r + s * g$
 - et ainsi à obtenir la signature (z, y)
 - VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste à :
 - calculer $g = G(m, z)$;
 - calculer $x = a^y * a^{(-n * g)} \text{ modulo } n$;
 - calculer $h = H(x)$;
 - vérifier que $z^2 = h \text{ modulo } n$ et que y est dans le bon intervalle $[0, R + 2^{(||s|| + ||g||)}]$.
8. Schéma de signature selon la revendication 4 caractérisé en ce qu'il utilise le schéma Poupard-Stern comme étant le procédé d'identification sans divulgation d'information, et le schéma Rabin, comme étant le schéma de signature, tel qu'il utilise les procédés suivants :
- GEN_INV est le procédé de génération de clé publique et clé secrète et consiste à :
 - générer un paramètre a appelé base ;
 - générer deux premiers p et q ;
 - retourner $(n = p * q, \alpha = a^{(-n)} \text{ modulo } n)$ comme clé publique, et $(p, q, s = n - (p - 1) * (q - 1))$ comme clé privée, la taille de s en bit étant connue, et notée $||s||$,

ainsi que celle de la fonction de hachage G , notée $||g||$, R désignant la valeur maximale du nombre aléatoire dans le procédé de génération de signature ;

- 5 • $SIGN_INV$ est le procédé de génération de signature, et consiste à :
 - o générer un nombre aléatoire r , entre 0 et R , et calculer $x = a^r \text{ modulo } n$
 - o pré-calculer $h = H(x)$
 - 10 o pré-calculer, à l'aide de p et q , la signature z comme étant une racine carrée de $h \text{ modulo } n$
 - o calculer $g = G(m, z)$, où m est le message à signer, une fois celui-ci connu ;
 - 15 o calculer $y = r + s * g$
 - et ainsi à obtenir la signature (z, y)
- VER_INV est le procédé de vérification de signature (z, y) du message m , et consiste :
 - calculer $g = G(m, z)$;
 - 20 ▪ calculer $x = a^y * \alpha^g \text{ modulo } n$;
 - calculer $h = H(x)$, et enfin
 - vérifier que $z^2 = h \text{ modulo } n$ et que y est dans le bon intervalle $[0, R + 2^{(||s|| + ||g||)}]$.

25

9. Schéma de signature selon la revendication 5 ou 6, caractérisé en ce que la clé privée est générée sous la forme :

($d = e^{-1} \text{ modulo } (p-1)*(q-1)$, $k = (e*d-1)/((p-1)*(q-1))$),
 30 et dans lequel, au moment de la signature, s est recalculé par la formule $s = n - (ed-1)/k$.

10. Schéma de signature selon la revendication 7 ou 8, caractérisé en ce que la clé privée est générée sous la forme : (p, q) et dans lequel, au moment de la signature, s est recalculé par la formule $s = p * q - (p-1) * (q-1)$.

11. Schéma de signature selon la revendication 4, caractérisé en ce que le procédé d'identification sans divulgation d'information est un des procédés suivants : Fiat-Shamir, Poupard-Stern, Girault-Poupard-Stern, et en ce que le schéma de signature est un des schémas de signature suivants : RSA, Rabin, Paillier, El Gamal, Goh-Jarecki.

12. Schéma de signature selon l'une des revendications revendication 5 ou 6, dans lequel d est défini comme : $d = e^{-1}$ modulo $\text{PPCM}(p-1, q-1)$.

13. Schéma de signature selon l'une des revendications revendication 5 ou 6, dans lequel d est défini comme : $d = e^{-1}$ modulo $u * \text{PPCM}(p-1, q-1)$, où u est un nombre entier positif non nul.

14. Schéma de signature à sécurité prouvé « tight » et « à la volée » destiné à un composant électronique selon l'une quelconque des revendications précédentes caractérisé en ce que le schéma de signature s'effectue sur un composant électronique portable.

15. Schéma de signature selon la revendication 14 caractérisé en ce que le composant électronique portable est une carte à puce.

- 5 16. Schéma de signature l'une quelconque des revendications 1 à 3, caractérisé en ce que le calcul s'effectue sur un module TPM pour « Trusted Platform Module » installé dans un PC non sécurisé.