(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 811 173**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **01 00781**

(51) Int Cl[7] : **H 04 L 9/30**

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 22.01.01.

(30) **Priorité :** 28.06.00 FR 00008307.

(43) **Date de mise à la disposition du public de la demande :** 04.01.02 Bulletin 02/01.

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

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

(71) **Demandeur(s) :** *GENESTE JEAN FRANCOIS* — FR.

(72) **Inventeur(s) :** GENESTE JEAN FRANCOIS.

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

(74) **Mandataire(s) :**

(54) PROCEDES CRYPTOGRAPHIQUES A CLE PUBLIQUE BASES SUR LA DIFFICULTE DE TROUVER LES VALEURS PROPRES D'UN ENDOMORPHISME D'UN MODULE SUR UN ANNEAU OU UNE ALGEBRE QUELCONQUE.

(57) La présente invention concerne un procédé de cryptographie à clé publique, applicable à des données enregistrées sous forme de bits sur un support exploitable par une entité de calcul, lequel procédé, selon la façon dont il est utilisé peut fournir un procédé de chiffrement, un procédé de preuve d'identité zero-knowledge au sens de Fiat-Shamir, un procédé de signature de signature électronique ou un procédé réalisant une fonction dite à sens unique.

Ce procédé de cryptographie à clé publique de données enregistrées sous forme de bits sur un support exploitable par une ou plusieurs unités de calcul aptes à traiter des données d'entrée x pour fournir des données de sortie x', comportant au moins une étape de traitement de données, se caractérise par le fait que la ou les unités de calcul manipulent des matrices carrées à coefficients dans un anneau ou une algèbre, commutatifs ou non, associatifs ou non, les données d'entrée étant considérées comme des vecteurs appartenant à des modules sur ces anneaux ou algèbres, la manipulation binaire amenant en sortie éventuellement une matrice carrée ou un vecteur ou un scalaire ou deux options parmi les trois ou les trois à la fois, les coefficients restant toujours sur les mêmes anneaux ou algèbres.

FR 2 811 173 - A1

Procédés cryptographiques à clé publique basés sur la difficulté de trouver les valeurs propres d'un endomorphisme d'un module sur un anneau ou une algèbre quelconque

La présente invention concerne un procédé de cryptographie à clé publique, applicable à des données enregistrées sous forme de bits sur un support exploitable par une entité de calcul, lequel procédé, selon la façon dont il est utilisé peut fournir un procédé de chiffrement, un procédé de preuve d'identité zero-knowledge au sens de Fiat-Shamir, un procédé de signature de signature électronique ou un procédé réalisant une fonction dite à sens unique.

Depuis le milieu des années 1970 de nombreux systèmes de cryptographie à clé publique sont apparus, notamment le RSA de Rivest Shamir et Adelman ainsi que le système d'échange publique de clé dit de Diffie-Hellmann. D'autres systèmes furent proposés, la grande majorité basée sur le problème de la factorisation. Peu sont basés sur d'autres problèmes mais je citerai ici ceux qui sont basés sur les courbes elliptiques ou hyper-elliptiques. Néanmoins, dans tous les cas, les systèmes suggérés n'ont que peu à voir avec la factorisation et leur solidité semble dépendre davantage du fait que l'on ne connaît pas ou peu le domaine concerné (e.g. les courbes elliptiques) que du fait que le problème est réellement difficile à résoudre. On est ainsi en présence d'un phénomène préoccupant. Soit on utilise un système cryptographique à clé publique basé sur la factorisation, soit on en utilise un basé sur un problème peu connu mais peut être pas très difficile. Alors pourquoi ne pas utiliser ceux basés sur la factorisation ? En fait, une personne qui aurait trouvé un algorithme en temps polynômial déterministe pour factoriser les grands nombres aurait-elle intérêt à publier sa découverte? Un pays comme les USA qui cherche à écouter toutes les communications de la terre via la NSA (National Security Agency) n'aurait-il pas intérêt à laisser croire que le problème de la factorisation est difficile de façon à, le cas échéant, continuer à écouter, sans être soupçonné, le monde entier ?

Dans ce qui suit, je vais présenter un procédé cryptographique à clé publique qui est, en général, bien plus compliqué à "casser" que de résoudre le problème de la factorisation. Cependant, dans certains cas, identifiés, le problème en question sera équivalent à la factorisation. L'utilisateur n'aura qu'à choisir ce qu'il veut. Je montrerai ensuite que selon comment on s'en sert on peut obtenir un procédé de preuve interactive zero-knowledge à

résultat indistingable, un procédé de signature électronique et un procédé de construction d'une fonction à sens unique. Enfin, les procédés présentés ici étant parallélisables, je montrerai comment réaliser des circuits intégrés parallèles qui les réalisent et permettent de faire du chiffrement à clé publique en temps réel, ce que n'arrive à faire aucun autre procédé de ce type à l'heure actuelle.

Les calculs explicités ici sont censés être faits sur des données enregistrées sous la forme de bits, et exécutés par des entités de calcul du type "ordinateur" ou selon le cas, des circuits électroniques spécifiques. Le chiffrement consistera à traiter les données d'entrée x pour fournir une sortie x', l'unité de calcul effectuant un certain nombre d'étapes de traitement de données, ce traitement utilisant un certain nombre de fonctions, notamment matricielles. Pour les preuves d'identité zero-knowledge ou les signatures électroniques, le même procédé sera utilisé, mais il fera intervenir des interactions entre 2 entités de calcul qui traiteront des données enregistrées elles aussi sous forme binaire.

■ 1. Notations - Rappels

Soit A un anneau quelconque et E un A-module de type fini. Soit f un endomorphisme de E. On dira qu'un vecteur $x \in E^*$ est un vecteur propre de f si $\exists\lambda\in A$ / $f(x)=\lambda x$. Pour plus de simplicité on ne considérera ici que les modules à gauche, mais les résultats décrits sont les mêmes et sont aussi valables pour les modules à droite. De même, λ est une valeur propre de l'endomorphisme f si $\exists x \in E^*$/ $f(x)=\lambda x$. Dans le cas où A est un anneau commutatif, une façon de calculer les valeurs propres de f est de résoudre son polynôme caractéristique.

Lorsqu'un tirage sera fait au hasard cela voudra dire qu'il est fait avec la distribution uniforme sur l'ensemble considéré. Les valeurs propres seront, en général, tirées dans le centre de l'anneau. Si y et z sont deux vecteurs de E, alors $\xi(y,z)$ est une fonction à sens unique depuis un sous-ensemble des bits des 2 dernières coordonnées de y et z dans A, ou bien, selon le cas, $\xi(\alpha)$ sera une fonction à sens unique de A dans A. Par exemple, dans certains cas, la fonction $\xi(y,z)$ pourra être l'élévation au carré dans A de la deuxième coordonnée de y.

Dans la suite, lorsque l'on dira "Bob" ou "Alice", il faudra entendre la machine de Bob ou Alice. Cette personnalisation des entités de calcul nous paraît permettre une meilleure compréhension.

■ 2. Description en dimension 3

On se place dans un contexte où Bob veut envoyer un message à Alice. E est un A-module de dimension 3, et on suppose que les messages à envoyer sont x_1 pour le premier jusqu'à x_m pour le dernier. Pour fabriquer sa clé publique Alice tire au hasard 3 valeurs

distinctes non nulles dans A. Soient λ_1, λ_2 et λ_3 ces valeurs. Alice forme alors la matrice

$F_0 = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$, puis elle tire au hasard une matrice de la forme $h^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \rho & \varphi \\ 0 & \omega & \psi \end{pmatrix}$ qui soit inversible. Elle forme alors la matrice $F = h^{-1} F_0 h$ qui est la clé publique. Dans la suite cette matrice sera écrite

$F = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \alpha & \beta \\ 0 & \gamma & \delta \end{pmatrix}$. Cette clé publique est enregistrée dans un annuaire électronique "hardware" consultable et "importable" depuis l'extérieur par une connexion, par exemple, du type Internet.

Algorithme de chiffrement du $i^{ème}$ message x_i

Etant donné x_i Bob tire au hasard 2 vecteurs y_i et z_i tels que $\{x_i, y_i, z_i\}$ forme une base de E.

Soit alors $g^{-1} = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$ la matrice

formée des coordonnées de ces vecteurs en colonne. Elle est évidemment inversible. Bob calcule alors $f = g^{-1}$ Fg et $\xi(y_i, z_i)x_i + y_i + z_i$. Bob envoie alors à Alice le couple $(f, \xi(y_i, z_i)x_i + y_i + z_i)$.

Algorithme de déchiffrement du $i^{ème}$ message

Comme Alice connaît les valeurs propres de f, Alice cherche 3 vecteurs propres de f associés à chacune des valeurs λ_i. Si on considère le vecteur propre $u_i = \begin{pmatrix} u_{i1} \\ u_{i2} \\ u_{i3} \end{pmatrix}$ associé à λ_2 par exemple, alors ce dernier est proportionnel à $g^{-1} \begin{pmatrix} 0 \\ \rho \\ \omega \end{pmatrix}$. On a donc le système d'équations

$$\begin{cases} \rho g_{12} + \omega g_{13} = k u_{i1} \\ \rho g_{22} + \omega g_{23} = k u_{i2} \\ \rho g_{32} + \omega g_{33} = k u_{i3} \end{cases}.$$

De même si $v_i = \begin{pmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \end{pmatrix}$ est le vecteur propre associé à λ_3, on a alors v_i qui est proportionnel

à $g^{-1}\begin{pmatrix}0\\ \varphi\\ \psi\end{pmatrix}$. On obtient alors le système suivant :

$\begin{cases}\varphi g_{12}+\psi g_{13}=k'v_{i1}\\ \varphi g_{22}+\psi g_{23}=k'v_{i2}\\ \varphi g_{32}+\psi g_{33}=k'v_{i3}\end{cases}$. On peut alors écrire ces équations sous la forme suivante.

$(g_{12}, g_{1,3})=(ku_1, k'v_1)\begin{pmatrix}\varrho & \psi\\ \omega & \varphi\end{pmatrix}^{-1}$, $(g_{22}, g_{2,3})=(ku_2, k'v_2)\begin{pmatrix}\varrho & \psi\\ \omega & \varphi\end{pmatrix}^{-1}$ et $(g_{32}, g_{3,3})=(ku_3, k'v_3)\begin{pmatrix}\varrho & \psi\\ \omega & \varphi\end{pmatrix}^{-1}$. Rappelons que $y=\begin{pmatrix}g_{12}\\ g_{22}\\ g_{32}\end{pmatrix}$ et $z=\begin{pmatrix}g_{13}\\ g_{23}\\ g_{33}\end{pmatrix}$. Si on écrit alors

$\begin{pmatrix}\varrho & \psi\\ \omega & \varphi\end{pmatrix}^{-1}=\begin{pmatrix}\varrho' & \psi'\\ \omega' & \varphi'\end{pmatrix}$

nos équations peuvent alors être écrites sous la forme $\begin{cases}y_i=\varrho' ku_i+\omega' k' v_i\\ z_i=\psi' ku_i+\varphi' k' v_i\end{cases}$. Ce qui prouve que y_i et z_i sont dans le plan engendré par les valeurs propres λ_2 et λ_3. Le texte chiffré est composé de la matrice f et d'un vecteur aléatoire $U=\xi(y_i, z_i)x_i+y_i+z_i$. Ecrivons alors que $U-(\varrho'+\psi')ku_i-(\omega'+\varphi')k'v_i$ est colinéaire à n'importe quel vecteur propre non nul associé à λ_1. Appelons ce dernier vecteur μ. On

obtient alors $\begin{cases}U_1-(\varrho'+\psi')\,ku_{i1}-(\omega'+\varphi')\,k'v_{i1}=\eta\mu_1\\ U_2-(\varrho'+\psi')\,ku_{i2}-(\omega'+\varphi')\,k'v_{i2}=\eta\mu_2\\ U_3-(\varrho'+\psi')\,ku_{i3}-(\omega'+\varphi')\,k'v_{i3}=\eta\mu_3\end{cases}$ où les inconnues sont k, k' et η. Ce système se résoud aisément et a une solution unique qui va donner k et k' et donc va permettre de retrouver σ. Le reste du déchiffrement est alors trivial.

■ 3. Difficulté de casser le système

La difficulté de casser le système précédent vient de la difficulté de trouver les valeurs propres d'une matrice à coefficients dans un anneau quelconque. On peut en particulier démontrer que lorque l'anneau est $\mathbb{Z}/n\mathbb{Z}$ où n=pq est le produit de 2 grands nombres premiers comme dans le RSA, l'algorithme précédent est aussi dur à casser qu'il est difficile de factoriser le nombre n. Dans ce cas particulier, une fonction intéressante pour ξ est l'élévation au carré de la 2ième coordonnée de y modulo n.

■ 4. Preuve zero-knowledge

Considérons R un anneau non commutatif. Supposons que R vérifie la condition dite de Ore à droite. C'est à dire que $\forall$ a, b $\in$ R, $\exists$ a', b' $\in$ R / aa' = bb'. On démontre alors que sous cette condition, l'anneau R admet un anneau de fractions, c'est à dire des éléments du type a/b où a $\in$ R et b $\in$ S où S est une partie de R sur laquelle on peut construire une

relation d'équivalence. Dans ce cadre, pour peu que a ∈ S, l'inverse de a/b s'écrit b/a. La construction d'un tel anneau de fraction est unique dès qu'on a choisi S. Cette construction ne sera donc pas décrite ici. Par contre, une même construction existe pour une condition dite de Ore à gauche avec le même résultat, mais une autre structure d'anneau. Le résultat présenté ici est donc valable pour les 2 cas. Ce qui est important, c'est que l'on travaille sur des classes d'équivalence et non sur des objets numériques directement. Appelons alors C_1 le procédé de chiffrement du paragraphe 2 et D_1 le procédé de déchiffrement du paragraphe 2. Dans le scenario présent, Alice veut prouver son identité à Bob. De même que dans les protocoles zero-knowledge classiques, elle va montrer qu'elle connaît une information connue d'elle seule, sans donner davantage d'information que le fait qu'elle connaît cette information. En clair, Alice va montrer qu'elle connaît les valeurs propres de F. Le protocole se déroule comme suit : Bob chiffre un message tiré au hasard dans le module sur l'anneau A des fractions de R avec C_1 et l'envoie à Alice. A la réception, Alice vérifie qu'il est de la forme (f, v) où f est une matrice 3×3 à coefficients dans A et v un vecteur de dimension 3 sur A. Si tel n'est pas le cas, alors Alice arrête le protocole. Sinon Alice déchiffre avec D_1. On montre facilement que le message obtenu est x' si le message de Bob est x et que chaque coordonnée de x' est dans la meme classe d'équivalence que celle correspondante de x. Les deux vecteurs sont donc équivalents et, d'un point de vue théorique on a déchiffré le message. Néanmoins, d'un point de vue pratique, on a, numériquement, 2 vecteurs équivalents potentiellement différents. Il n'y a aucune raison théorique ou pratique pour favoriser une forme plutôt qu'une autre. Par contre, seul le détenteur des valeurs propres peut trouver un équivalent de x. A la réception de x' Bob vérifie l'équivalence des 2 vecteurs. Si tel n'est pas le cas, Bob refuse d'identifier Alice, sinon Bob accepte.

On démontre qu'un tel protocole est un protocole de preuve interactive Zero-knowledge au sens de Fiat-Shamir à résultat inconditionnelement indistingable.

■ 5. Signature

Nous allons nous inspirer ici d'une technique due à Jean-Jacques Quisquater et Louis Guillou qui permet, à partir d'un système interactif de preuve zero-knowledge, de déduire un système de signature interactif. Comme dans le paragraphe 2, Alice construit sa matrice publique F. On suppose ici que le message à signer est m ∈ E. Comme dans le procédé C_1, Bob tire au hasard 3 vecteurs tels que {x, y, z} forme une base de E et on appelle g^{-1} la matrice formée de ces vecteurs en colonne, ainsi que f=g^{-1} Fg. f est envoyée à Alice. A la réception, Alice calcule comme dans le paragraphe 2, x, y et z ainsi que f(m) et le décompose sur la base {x, y, z}. Elle a ainsi f(m)=$t_1 + t_2 + t_3$. Alice tire alors au hasard 3 valeurs α, β et γ et envoie à Bob le couple ($\alpha t_1 + \beta t_2 + \gamma t_3$, $\xi(\alpha)$). Bob décompose f(m) sur la base {x, y, z}, obtient les t_i , décompose $\alpha t_1 + \beta t_2 + \gamma t_3$sur la base des t_i, ce qui lui donne α, β et γ et calcule $\xi(\alpha)$ pour vérifier que la deuxième partie du couple correspond bien.

On remarquera que cet algorithme marche quelque soit l'anneau considéré, commutatif ou non, anneau des fractions ou non.

■ 6. Le choix de l'anneau

Dans les sections précédentes nous avons décrit des procédés qui marchent dans quasiment tous les types d'anneaux. Cependant, on peut ici donner quelques conseils judicieux et apporter des précisions. Tout d'abord, les procédés C_1 et D_1 marchent dans tout anneau. Lorsque l'anneau est commutatif, il faut vérifier que le problème de trouver les valeurs propres est un problème difficile. A priori, mais de manière non exhaustive, un anneau local pourra marcher. Une façon de construire un tel anneau est de choisir dans un anneau R un idéal premier P. Soit S = R - P. Alors $A = S^{-1} R$ est un anneau local. Pour le cas des preuves zero-knowledge et des signatures, l'anneau quotient d'un anneau non commutatif peut être choisi. Là encore, même si la construction est un peu différente, on pourra choisir un anneau local. Enfin, on n'est pas obligé d'être réellement dans un anneau. En effet, l'associativité de la multiplication n'est pas une condition nécessaire. Ainsi un anneau convenable pourrait être le suivant. Soit $R=\mathbb{O}_n$ l'ensemble des octonions modulo n où n=pq est le produit de 2 nombres premiers. Soit S l'ensemble des éléments de R dont aucune des composantes n'est divisible par p. Considérons alors l'anneau non associatif des fractions $A = S^{-1} R$. Les algorithmes précédents marchent dans un tel anneau.

Tout cela n'est donné qu'à titre d'exemple, les procédés décrits marchant dans quasiment toutes les structures à 2 opérations.

■ 7. Description en dimension 4

Les procédés de la dimension 3 peuvent être adaptés à la dimension 4. Nous reprenons les mêmes notations que précédemment. Soit $F_0 = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix}$ la clé secrète d'Alice où les λ_i sont choisies au hasard non nulles et toutes distinctes. Alice tire alors au hasard une

matrice $h^{-1} = \begin{pmatrix} \sigma & \tau & 0 & 0 \\ \mu & \nu & 0 & 0 \\ 0 & 0 & \rho & \varphi \\ 0 & 0 & \omega & \psi \end{pmatrix}$ et publie la matrice $F = h^{-1} F_0 h$. Un message est toujours un vecteur u ∈ E, E qui est cette fois-ci de

dimension 4 sur A.

Chiffrement d'un message

Bob veut envoyer u. Il tire alors au hasard x et pose y = u - x. Il complète en choisissant au hasard z et t tels que {x, y, z, t} forme une base de E. Appelons $g^{-1} = (g_{i,j})$ la matrice formée des colonnes de ces vecteurs. Bob envoie alors à Alice f=g^{-1} Fg suivie d'un vecteur du type $\xi(z,t)(x + y) + z + t$ ou $\xi_1(z, t)x + \xi_2(z, t)y + z + t$.

Déchiffrement du message

On raisonne comme au paragraphe 2 pour retrouver les vecteurs z et t en écrivant que le vecteur propre associé à λ_3 est proportionnel

à $g^{-1}\begin{pmatrix} 0 \\ 0 \\ \rho \\ \omega \end{pmatrix}$ et en écrivant que celui associé à λ_4 est proportionnel à $g^{-1}\begin{pmatrix} 0 \\ 0 \\ \varphi \\ \psi \end{pmatrix}$. En résolvant les systèmes d'équations on obtient z et t et donc $\xi(z, t)$ ou les $\xi_i(z, t)$ et donc u de manière évidente.

Propriété particulière

A partir de la dimension 4 on peut démontrer que le système résiste à une attaque adaptative à chiffré choisi, alors que dans le cas de la dimension 3 l'attaque ne doit pas être adaptative.

Généralisations

Un tel schéma ou celui en dimension 3 peuvent être facilement généralisés à des dimensions supérieures à 4. Cela étant immédiat, nous n'irons pas plus loin dans la description de tels systèmes. Cependant, nous citerons pour mémoire le cas des dimensions supérieures ou égales à 5 pour lesquelles, en plus de la difficulté traditionnelle de trouver des valeurs propres vient s'ajouter, dans le cas d'un anneau commutatif en particulier, l'impossibilité de la résolution par radicaux du polynôme caractéristique.

■ 8. Variantes

Nous présentons ici, de manière non exhaustive, quelques variantes des procédés précédents de façon à prouver la puissance du système. Ces variantes sont toutes en dimension 3 ou 4 mais se généralisent à d'autres dimensions très facilement. Nous ne donnerons ici que des descriptions partielles, le reste étant identique aux paragraphes précédents. De plus, sauf mention du contraire, nous nous intéresserons principalement aux aspects de chiffrement.

Variante numéro 1

Dans le procédé C_1, au lieu d'envoyer à Alice (f, $\xi(y, z)x + y + z$), Bob envoie

directement (f, x+y+z). On démontre qu'aucune sécurité n'est apportée par un tel procédé s'il est sur un anneau commutatif. Par contre, dès que l'anneau est non commutatif, la sécurité est assurée (via une conjecture qui est hors du cadre de ce texte). Ainsi le chiffrement et déchiffrement sont simplifiés d'autant pour un anneau non commutatif. De même, cette variante peut être étendue à la preuve zero-knowledge du paragraphe 4 et à la signature du paragraphe 5.

Variante numéro 2

On se place ici en dimension 4 et strictement dans le cas d'un anneau **non commutatif**. Sous la condition que certains éléments de la matrice g^{-1} soient connus et par exemple égaux à 1, on peut alors modifier le chiffrement comme suit. Supposons que l'équivalent d'une ligne de g soit connu, par exemple $g_{4,1} = g_{4,2} = g_{1,3} = g_{1,4} = 1$, alors il suffit à Bob d'envoyer la matrice f. En suivant le même type de raisonnement que pour D_1 Alice retrouve les vecteurs x, y, z et t et peut donc calculer x+y et z+t qui forme alors le message en clair.

Variante numéro 3

Nous allons donner ici un système de preuve interactive zero-knowledge en dimension 2 ; il nous faut pour cela une hypothèse supplémentaire, celle selon laquelle Alice sait trouver le valeurs propres d'une matrice 2×2 en temps polynomial déterministe. Soit $F_0 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ tirée au hasard par Bob. Ce dernier tire alors au hasard 2 vecteurs x et y formant une base de E et soit g^{-1} la matrice formée de ces vecteurs en colonne. Soit $F=g^{-1} F_0 g$. Bob envoie alors à Alice le couple (F, x+y). Alice, qui sait retrouver les valeurs propres peut alors retrouver x et y et elle tire au hasard α et β, puis envoie à Bob (αx+βy, $\xi(\alpha)$). A la réception, Bob, qui connaît la base de vecteurs propres {x, y} retrouve α et β et calcule $\xi(\alpha)$ pour voir la correspondance avec le deuxième élément du couple qu'Alice lui a envoyé. Si tel est le cas, Bob accepte, sinon Bob refuse.

Un tel algorithme est particulièrement bien adapté au cas où A=$\mathbb{Z}$/n$\mathbb{Z}$ et n=pq comme dans le RSA. En effet, connaissant la factorisation de n, Alice peut facilement résoudre le polynôme caractéristique en utilisant, entre autre, le théorème chinois des restes et dans ce cas on peut par exemple choisir $\xi(\alpha) = \alpha^2 \bmod n$.

Variante numéro 4

Nous donnons ici, en dimension 3 une version de preuve interactive zero-knowledge qui marche dans un anneau commutatif. De même que dans le paragraphe 2, Bob chiffre un message tiré au hasard, x, à l'aide de C_1. Alice le déchiffre et obtient donc x, y et z. Elle renvoie donc à Bob (αx+βy+γz,$\xi(\alpha)$) et Bob vérifie alors comme dans la variante numéro 3. Des schémas similaires marchent dans toutes les dimensions.

Nota

Les variantes 3 et 4 donnent bien sûr naissance aux systèmes de signature interactifs

du type Guillou-Quisquater.

Variante numéro 5

Voici un système de chiffrement basé sur la difficulté de trouver les valeurs propres d'un endomorphisme lorsqu'Alice connaît une information qui lui permet elle, de les calculer. C'est le cas typique où on travaille dans l'ensemble des entiers modulo n et où seule Alice connaît la factorisation de n. Bob, lorsqu'il veut s'adresser à Alice, tire au hasard une matrice diagonale $F=\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$ et la garde sans modification tant qu'il correspond avec Alice. Le $i^{ème}$ message est un vecteur $x_i \in E$ et Bob tire au hasard y_i et z_i pour que les 3 vecteurs forment une base de E. Soit alors g_i^{-1} la matrice formée de ces vecteurs en colonne. Bob envoie à Alice $(f_i = g_i^{-1} F g_i, \xi(y_i, z_i)x_i + y_i + z_i)$. Pour déchiffrer, Alice calcule les valeurs propres de f_1 (seulement pour le premier message, qui au passage peut être public pour faciliter encore les choses), et cherche une base propre de f_i. Elle décompose ensuite $\xi(y_i, z_i)x_i + y_i + z_i$ sur cette base, obtient y_i et z_i donc $\xi(x_i, y_i)$ et par voie de conséquence x_i.

Variante numéro 6

Nous présentons ici une généralisation directe de l'algorithme présenté au paragraphe 2. Si n est la dimension de E, on considère une matrice n×n, clé publique d'Alice et de la forme $F=\begin{pmatrix} \lambda_1 & 0 \\ 0 & R \end{pmatrix}$ ou R est une matrice non diagonale comme dans le paragraphe 2 mais de taille (n-1)×(n-1) et qui a pour valeurs propres $\lambda_2, \ldots, \lambda_n$. Le message est toujours un vecteur de E et Bob envoie à Alice le message chiffré ($f=g^{-1} F g$, $\xi(y, z, t, \ldots)x+y+z+t+\ldots$). Le déchiffrement est alors évident compte tenu de ce qui précède.

■ 9. La parallélisation

Dans ce paragraphe, nous allons voir deux types de parallélisation.

■ 9.1. Parallélisation brute

Ici, les opérations sont supposées câblées dans du hardware. Nous avons alors plusieurs briques de hardware pour réaliser les opérations. Le but est de diminuer au maximum le nombre de multiplications série que l'on va avoir à faire. Les opérations de base qui sont câblées sont la multiplication, l'addition, la soustraction et l'addition dans l'anneau A. Elle sont représentées schématiquement sur les figures 1 à 4. On peut alors définir la multiplication parallèle de 2 matrices. Prenons l'exemple en dimension 3. Si $P=(P_{i,j})$ et si $M=(M_{i,j})$ alors on a $PM = (\alpha_{i,j})$ où $\alpha_{i,j} = \sum_{k=1}^{3} M_{k,j} P_{i,k}$. On remarquera que cette formule est valable que l'anneau soit commutatif ou non. La formule de multiplication des matrices sera indiquée selon le schéma figure 5 et le détail effectué est selon la figure 6. Sur cette

dernière figure on peut constater que la multiplication de 2 matrices entre elles peut se faire en le temps d'une multiplication série, puisque pour i et j fixés on peut effectuer les 3 multiplications en parallèle. Conformément à la coutume, on néglige le temps de calcul des additions et soustractions. On pourrait continuer ainsi et détailler chaque opération nécessaire à l'exécution de chaque procédé décrit ci-dessus. La chose étant évidente, nous n'irons pas plus loin, mais il est aisé de concevoir un circuit électronique spécifique à chaque procédé qui effectue en parallèle le maximum possible d'opérations de façon à diminuer au maximum le temps de calcul vu par l'utilisateur. Ainsi, dans un anneau commutatif, le temps nécessaire, avec un tel circuit, pour calculer C_1 est le temps de 5 multiplications séries. Pour cette évaluation, nous avons considéré que le temps d'une division est équivalent à celui d'une multiplication, ce qui est courant mais peut être optimiste. A l'inverse, dans un anneau de fraction, le temps d'une division est négligeable. En ce qui concerne D_1, 7 multiplications série sont nécéssaires.

Dans les anneaux de fractions non commutatifs, ces temps peuvent varier quelque peu selon l'algorithme d'inversion choisi pour une inversion de matrice. Cependant, comme le temps d'une division est négligeable, on arrive là encore, à d'excellentes performances.

■ 9.2. Parallélisaton par microprocesseurs

La parallélisation précédente a l'avantage de procurer les meilleurs débits possibles. Elle a cependant l'inconvénient de ne pas utiliser toute la puissance des procédés présentés ici. En effet, dans le paragraphe 9.1., une fois l'anneau choisi, on ne peut plus en changer. Comme nous l'avons dit plus haut, en introduction, pour le RSA par exemple, le jour où le problème de la factorisation est résolu, si c'est possible, tous les circuits intégrés qui effectuent cet algorithme sont bons à jeter à la poubelle. Pire, si l'on doute (c.f. ce qui a été dit sur la NSA) on est prisonnier du système et il n'y a guère d'alternative. Nous proposons donc ici une parallélisation d'un type nouveau.

Le lecteur est invité à reprendre les figure 1 à 4 précédentes, et à considérer que chaque "boîte" est un microprocesseur pour lequel on peut programmer une opération dans l'anneau de son choix, et cela dans un langage évolué du type *Mathematica*. On peut alors créer des macro microprocesseurs comme celui de la figure 5 en parallélisant l'utilisation des premiers dans une structure du type de celui de la figure 6. C'est uniquement le compilateur qui insérera dans chaque partie adéquate le "morceau" d'algorithme convenable. En continuant ainsi, toujours avec le souci de la parallélisation, les procédés de chiffrement et déchiffrement se construisent comme des sortes de superprocesseurs.

L'avantage de cette technique réside dans le fait que si on a un doute sur un anneau, on peut en changer immédiatement. Le problème est alors basé sur le cas très général de la recherche des valeurs propres d'un endomorphisme sur un module sur un anneau quelconque, voire, comme nous l'avons dit pour les octonions, sur une algèbre quelconque qui peut ne pas être associative. Ce problème, qui englobe celui de la factorisation, est très large et bien plus complexe que cette dernière.

■ 10. Une nouvelle fonction à sens unique

Les fonctions à sens unique sont d'une importance capitale en tant que primitives cryptographiques. Dans ce cadre, il est toujours interessant d'en avoir de nouvelles à disposition, puisque certains protocoles formels existent et sont bâtis sur la conjecture selon laquelle de telles fonctions existent.

Nous reprenons donc le procédé C_1 mais dans un anneau de fractions non commutatif, où tous les éléments de g sont tirés au hasard et les valeurs propres choisies avec la distribution uniforme sur tout l'anneau et sont distinctes. Un tel procédé, C_1', est alors une fonction à sens unique qui à x associe une image. La difficulté de retrouver x réside dans le fait que dans un tel anneau, en général, la recherche d'un vecteur propre est très difficile (e.g. l'ensemble des vecteurs propres peut ne pas être un sous-module du module E).

Bien entendu, cette fonction à sens unique peut se généraliser à n'importe quelle dimension.

Revendications

1. Procédé de cryptographie à clé publique de données enregistrées sous forme de bits sur un support exploitable par une ou plusieurs unités de calcul aptes à traiter des données d'entrée x pour fournir des données de sortie x', comportant au moins une étape de traitement de données, caractérisé par le fait que la ou les unités de calcul manipulent des matrices carrées à coefficients dans un anneau ou une algèbre, commutatifs ou non, associatifs ou non, les données d'entrée étant considérées commes des vecteurs appartenant à des modules sur ces anneaux ou algèbres, la manipulation binaire amenant en sortie éventuellement une matrice carrée ou un vecteur ou un scalaire ou deux options parmi les trois ou les trois à la fois, les coefficients restant toujours sur les mêmes anneaux ou algèbres.

2. Procédé cryptographique selon la revendication 1 caractérisé en ce que la matrice F constituant la clé publique est carrée et diagonalisable, en ce que la fabrication de la clé publique se fait en tirant une matrice à coéfficients dans l'algèbre sur laquelle on travaille, avec la distribution uniforme, et en ce que cette matrice doit être inversible.

3. Procédé cryptographique selon les revendications 1 et 2 caractérisé en ce que dans un mode de réalisation, le chiffrement du vecteur qui représente le message en clair constitue en la complétion de distribution uniforme (sur un sous-ensemble du module) de ce vecteur en base dans le module de travail, cette complétion étant "archivée" dans l'unité de calcul sous la forme d'une matrice carrée, chaque colonne de cette matrice étant constituée d'un des vecteurs de la base, le premier étant le message en clair, la continuation du chiffrement se faisant en changeant la matrice constituant la clé publique via un changement de base lié à la matrice de complétion ci-dessus évoquée, le résultat étant accompagné du vecteur "message en clair" multiplié, éventuellement, par le résultat de l'application d'une fonction à sens unique sur les coordonnées des vecteurs complétant la base, additionné de ces vecteurs de complétion.

4. Procédé cryptographique selon l'une quelconque des revendications précédentes, permettant la preuve d'identité zero-knowledge, caractérisé en ce que lors de l'utilisation d'un anneau ou d'une algèbre de fractions non commutative, dans le mode de réalisation de la revendication 3, le prouveur fournit au vérifieur le résultat de son déchiffrement comme preuve de connaissance, à savoir un vecteur égal au message en clair qui aura au préalable été tiré au hasard avec la distribution uniforme, l'égalité étant établie dans l'anneau ou l'algèbre de fractions non commutative concernée, mais en fait potentiellement un vecteur numériquement différent mais équivalent au message en clair, équivalence au sens de celle définie par la condition de Ore.

5. Procédé cryptographique selon les revendication 1, 2 et 3, permettant la signature électronique, caractérisé en ce qu'une fois la matrice $f=g^{-1}$ Fg ayant été envoyée au "fabricant" de la clé publique F, ce dernier calcule le changement de base effectué sur F, décompose le message à signer sur la base du module ainsi obtenue et envoie une combinaison linaire de cette décomposition accompagnée de l'image par une fonction à sens

unique de l'un des coefficients de la combinaison linéaire.

6. procédé cryptographique selon l'une quelconque des revendications précédentes, caractérisé en ce qu'un mode de réalisation met en oeuvre une parallélisation des calculs dans un circuit spécifique, ce circuit étant programmable ou non et donc, selon le cas, permettant le choix ou non, respectivement, de la structure d'anneau ou d'algèbre, retenue pour travailler.

figure 1

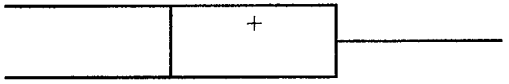

figure 2

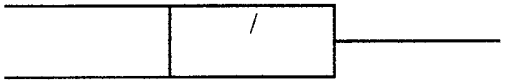

figure 3

figure 4

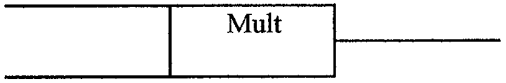

figure 5

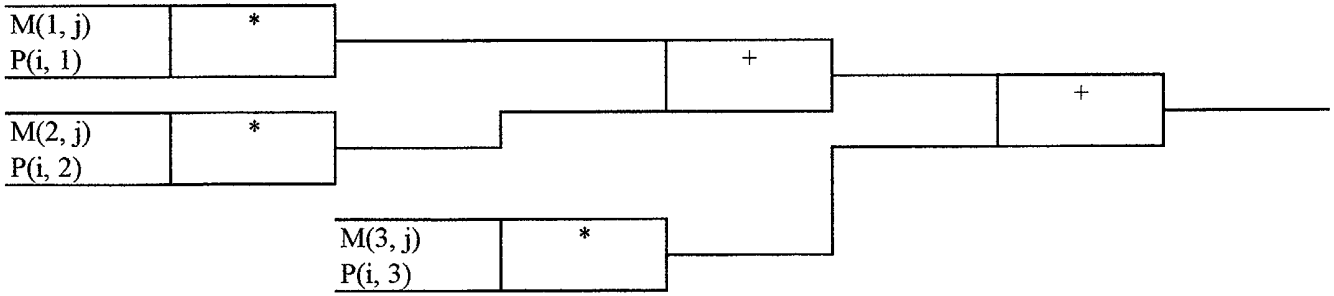

figure 6