Crypto Système à Clé Publique de McEliece basé sur les Codes Cycliques de Hamming

Naima Hadj-Said, Adda Ali-Pacha, A. Belgoraf, A. M'Hamed

Crypto Système à Clé Publique de McEliece basé sur les Codes Cycliques de Hamming

Naima HADJ-SAID[1] – Adda ALI-PACHA[1] – A. BELGORAF[1] – A. M'HAMED[2]

[1]Université des Sciences et de la Technologie d'Oran, BP 1505 El M'Naouer Oran 31036 ALGERIE
[2]Institut National des Télécommunications Evry- Paris
Tél./Fax : 213 /41 / 462 685 E.Mail : nim_hadj@yahoo.fr

Projet ANDRS*: Traitement des données : application à la radiothérapie CHU d'ORAN* (Code04/04/01/01/086)

Résumé: Le développement rapide des réseaux mondiaux et les immenses possibilités offertes par les transactions électroniques en communication continues, posent aujourd'hui de manière cruciale le problème de la protection de l'informations contre les erreurs de transmission d'une part, et d'autre part il faut que ces données soit non intelligibles sauf à l'auditoire voulu. Afin de pallier à ces deux contraintes on utilise le codage de l'information pour combattre les erreurs de transmissions et, le chiffrement des données est souvent utilisé pour lutter contre tout système d'espionnage.
Dans cette communication, on essaye d'introduire un crypto système à clef publique utilisant des codes correcteurs d'erreurs (c'est un système deux en un). Le système étudié est le crypto système de McEliece utilisant le code cycle de Hamming corrigeant une erreur simple.

Mots clés : Code de Hamming, cryptographie, chiffrement, clé publique, McEliece.

1. Introduction

Le problème majeur des dispositifs de télécommunication en plus de l'augmentation du débit de transmission qui peut être réglé par les méthodes de compression, est d'une part, les erreurs introduites par les supports de transmissions c.à.d. l'information reçue est erronée. Il est donc besoin de protéger cette information. Et d'autre part, le besoin croissant de sécuriser les données dans les domaines informatique et les télécommunications. Surtout maintenant avec la venue des réseaux téléinformatique, l'emploi des liaisons satellites et l'utilisation de l'Internet la situation a radicalement changé, dans la mesure où un même message transite par plusieurs machines avant d'atteindre son destinataire. A chaque étape, il peut être copié, perdu ou altéré. Le cryptage est donc nécessaire pour que les données soit non intelligibles sauf à l'auditoire voulu.

En 1976, avec l'invention du premier crypto système à clef publique par Diffie et Hellman [8]. L'idée nouvelle était de faire reposer la sécurité d'un système non pas sur la connaissance d'une clef (partagée secrètement par les utilisateurs), mais sur la difficulté d'inverser une fonction à sens unique avec trappe.
Une fonction à sens unique est simplement une fonction calculatoirement difficile à inverser.
Une trappe est un algorithme secret rendant facile cette inversion. Ainsi la trappe n'est connue que d'une personne, seule à pouvoir déchiffrer les messages créés en utilisant la fonction à sens unique qui est elle publique.

Dès 1978, McEliece [6] a imaginé le premier et le plus célèbre des crypto systèmes à clef publique utilisant des codes correcteurs d'erreurs. Comme nous allons le voir dans cette communication, la théorie des codes contient elle aussi de multiples problèmes bien structurés et difficiles à résoudre, plus ou moins biens adaptés pour une utilisation en cryptographie.

Dans ce travail on va essayer d'introduire un crypto système à clef publique utilisant des codes correcteurs d'erreurs. Le système étudié est le crypto système de McEliece utilisant le code cycle de Hamming.

2. Codes Correcteurs d'Erreurs

La construction d'un mot de code comportant n bits est effectuée à partir de k bits du message source k-uplet binaires $U=(u_1,u_2,u_3,\ldots..u_k)$, appelé généralement message d'information, et de r bits de redondance. La méthode de codage la plus simple consiste à laisser inchangés les k bits d'information et à les reporter tels quels dans le mot de code en ajoutant les r (= n-k) bits de redondance $\{a_1,a_2,\ldots,a_r\}$, qui sont généralement appelés bits de contrôles, le vecteur ligne V^T appelé mot code::

$$V^T=[\, v_1\, v_2\, \ldots\, v_n] =[\, u_1,u_2,u_3,\ldots..u_k\, a_1,a_2,\ldots,a_r\,].$$

- Lorsque les bits de contrôle sont calculés uniquement à partir des bits d'information du bloc auquel ils appartiennent, le code C(n , k) est appelé code de bloc.
- Lorsque les bits de contrôle sont calculés à partir des bits d'information appartenant à plusieurs blocs, le code est dit convolutionnel ou récurrent.

Le code de hamming qui est étudié dans cette communication est de type linéaire cyclique :

1. Les codes linéaire ont la propriété que l'ensemble des mots codes forment un espace vectoriel.
2. Les codes cycliques ont la propriété que toute permutation circulaire d'un mot code est un mot code.

Le code de Hamming est très employé dans la pratique pour protéger les informations courtes (16, 32 ou 64 bits), il est très employé pour les opérations sur les mémoires d'ordinateurs mais peu utilisé en transmission, il présente aussi les avantages suivants :

- plus facile à mettre en œuvre en logique câblée.
- Se prête bien à une extension de longueur de l'information à coder.

2.1 Matrices Génératrice et de Contrôle

Dès qu'on a la possibilité de déterminer deux matrices M_1 et M_2 possédant **n** colonnes telles que :

$$[M_1][M_2]^T=0 \quad (1)$$

On a défini un code linéaire à **n** positions [3]. En effet, si la matrice M_1, a pour dimension (m x n) et la matrice M_2 (k x n), on obtient tous les mots du code en prémutipliant M_1 par tous les m-uplets X_i :

$$<X_i> [M_1] = <C_i> \qquad i= 0 \text{ à } 2^m-1$$

Mais, comme on peut toujours mettre la matrice M_1 sous sa forme canonique en échelon :

$$[G] = [I_m\ A] \qquad \text{On a}$$
$$<X_i> [G]= <\text{m position d'information, n-m position de contrôle}>$$

Mais, pour que l'égalité (1) soit toujours satisfaite, on remarquera qu'il faut que M_2, mise sous sa forme canonique en échelon soit égale à $[-A^T\ I_k]$.
Nous désignerons cette matrice par [H] et on vérifie que :

$$[G][H]^T=0 \quad \text{avec} \quad k+m=n$$

La matrice G est la matrice génératrice du code (n, m), la matrice H est sa matrice de contrôle.
Or, comme $[G][H]^T = 0$ entraîne $[H][G]^T=0$ à tout code (n, m) correspond un code (n, n-m) appelé dual premier, le rôle des matrices étant inversé.

La matrice de contrôle H joue un rôle capital dans la détection ou la correction des erreurs. En effet :

Si $<X_i> [G]=<C_i>$, alors obligatoirement $[H](C_i)=0$

Cette équation matricielle représente un système de **k** équations à **k** inconnues qui sont les valeurs à attribuer aux k positions de contrôle quand les valeurs de positions d'information sont connues. Tous les n-uplets C_i vérifiant ce système sont des mots du code. Les k équations représentées par $[H]<C_i>=0$ définissent les relations de contrôle, elles permettent de déceler et de corriger les erreurs.

2.2 Codes Aléatoires

Pour obtenir un code aléatoire il suffit de tirer une matrice génératrice aléatoire et de chercher son image. Bien sûr, une fois choisi, le code n'est plus aléatoire, mais de façon générale, un code construit de cette façon aura de bonnes propriétés en moyenne : il a en général une bonne distance minimale. Malheureusement pour un tel code il n'existe pas d'algorithme de décodage polynomial. Cette dernière catégorie n'est pas vraiment une famille de codes puisqu'il s'agit en fait de tous les codes construits sans structures particulières.

3. Code Cyclique de Hamming

Un code de Hamming est un code cyclique C(n,k) généré par un polynôme primitif g(x) de degré $m \geq 3$ [1,2,4]. Avec les caractéristiques:

1. Longueur du mot de code $n = 2^m-1$.
2. Nombre de bits de contrôle $m = n - k$.
3. Nombre de bits d'information $k = 2^m-m-1$.
4. $d_{min} = 3$, le code corrige t erreurs (une erreur simple) $t = [(d_{min} - 1)/2] = 1$

3.1 Principe de codage

Soit $U = (U_0, U_1, \ldots, U_{k-1})$ le k–uplet à coder auquel on associe le polynôme U(x). Le codage consiste à :

a) Pré multiplier le polynôme U(x) associé au k – uplet à coder par x^{n-k}.
b) Obtenir le reste D(x) de la division de $x^{n-k} * U(x)$ par le polynôme générateur g(x).
c) Additionner D(x) et $x^{n-k} * U(x)$ pour obtenir le mot de code $V(x) = D(x) + x^{n-k} * U(x)$.

3.2 Principe de Décodage

Le processus de décodage est divisé en deux étapes :

- Détection d'erreurs dans le mot reçu.
- Correction de ces erreurs dans le cas échéant.

Le syndrome S(x)de mot reçu R(x) est définit par :

$$S(x) = reste(x^{n-k} * R(x) / g(x)) \qquad (2)$$

Avec $R^{(i)}(x)$: le polynôme obtenu après le $i^{ième}$ décalage à droite du R(x). Le syndrome correspondant au $i^{ième}$ décalage cyclique de R(x) peut être calculer par :

$$S^{(i)}(x) = reste(x^{n-k} * R^{(i)}(x) / g(x)). \qquad (3)$$

$S^{(i)}(x)$ peut être calculée par [7] :

$$S^{(i)}(x) = \begin{cases} reste\,(xS^{(i-1)}(x)/g(x)) & si\ \ i>0 \\ S(x) & si\ \ i=0 \end{cases} \qquad (4)$$

De (3) on montre que les $S^{(i)}(x)$ peuvent être calculé à partir de S(x) en utilisant l'équation de récurrence et par conséquent on n'est pas obligé de réaliser des décalages sur R(x) pour les calculée. La formation du syndrome est :

$$S(x) = S^{(0)} + S^{(1)}x + S^{(2)}x^2 + ... + S^{(n-k-1)}x^{n-k-1} \qquad (5)$$

La détection se fait suivant la valeur de S(x), on peut dire s'il y a des erreurs ou non, pour cela on a deux cas :

- S(x) = 0, R(x) est un multiple de g(x) donc le mot reçu est considéré comme le mot émis.
- S(x) <> 0, R(x) n'est pas un mot de code i.e.

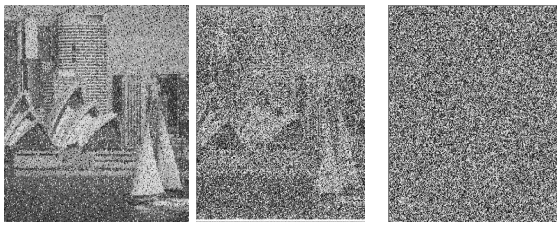
V(x) = R(x) +E(x), on va corriger les erreurs.

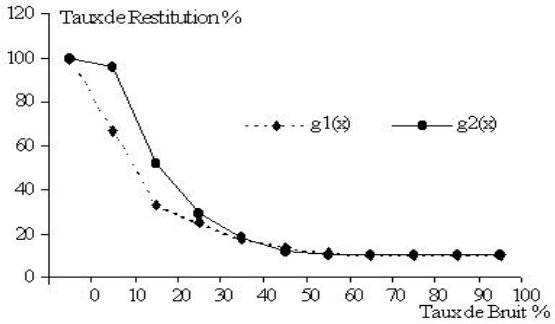
S(x) ne dépend que de la configuration d'erreurs E introduite et pas de V. En effet on a :

$$S(x) = reste(x^{n-k} * R(x)/g(x)) = reste(x^{n-k}E(x)/g(x)) \qquad (6)$$

3.3 Correction d'Erreurs

Pour interpréter les résultats de la correction d'erreurs,on prend comme image initiale Sydney BMP (256x256), le bruit utilisé est un Bruit Blanc Gaussien [5], et on varie les polynômes générateurs ($g_1(x)=1+x+x^4$, $g_2(x)=1+x+x^3$). Voici quelques images reçues pour différents valeurs de taux de bruit 10%, 30% et 80% pour le code de Hamming $g_1(x)=1+x+x^4$.





Graphe 1: TR en fonction du taux du bruit

A travers le graphe N° 1, on remarque que les taux de restitutions obtenu en utilisant le code de Hamming $g_2(x)$ sont meilleurs par rapport à ceux obtenu par le même code avec un polynôme générateur de degré égale à quatre. Le taux de redondance ou rendement (R=m/n) pour le premier code égale a $R_1=3/7$ est supérieur au deuxième $R_2=4/15$. Donc le taux de restitution est inversement proportionnel au rendement.

4. Crypto Système de McEliece

C'est le plus ancien crypto système à clef publique utilisant des codes correcteurs d'erreurs. Il a été imaginé par McEliece [6] en 1978, à peu près en même temps que RSA [7]. Comme tous les crypto systèmes à clef publique, ce système est constitué de 3 algorithmes :

1. la génération de clefs,
2. le chiffrement (utilisant la clef publique) et
3. le déchiffrement (utilisant la clef secrète).

McEliece a suggéré d'utiliser **les codes de Goppa**, qui sont des codes linéaires avec un algorithme rapide de décodage. On se propose de le faire avec le code cyclique de Hamming.

4.1 Génération de clef

On commence par générer un code de Hamming et sa matrice de parité G de taille k × n. On va mélanger cette matrice pour la rendre indistinguable d'une matrice aléatoire, pour cela on a besoin :

1. D'une matrice de permutation aléatoire P de taille n×n ayant un 1 dans chaque rangée et colonne et des 0 partout et,
2. D'une matrice inversible aléatoire S de taille k×k (S est une matrice ***brouilleur***).

La clef publique sera la matrice G' = S×G×P qui est indistinguable d'une matrice aléatoire. La sécurité de ce système repose sur le problème de distinguabilité du code de Hamming permuté d'un code aléatoire. La clef secrète est composée des trois matrices S, P et G qui permettent de retrouver la structure du code de Hamming et donnent donc accès à l'algorithme de décodage.

4.2 Chiffrement

Soit m un message de k bits que l'on veut chiffrer. On ne dispose pour cela que de la clef publique G'. On commence par calculer le mot de code C de longueur n associé à m :

$$C = m \times G' \qquad (7)$$

Ensuite on génère une erreur aléatoire e de longueur n. Le chiffré sera simplement le mot de code bruité :

$$C' = c + e \qquad (8)$$

4.3 Déchiffrement

Pour déchiffrer en connaissant P, S et G il suffit de calculer :

$$C' \times P^{-1} = mG'P^{-1} + eP^{-1} = m\,S \times G + eP^{-1} \qquad (9)$$

M S×G est un mot du code de Hamming et eP^{-1} est une erreur de poids t (car P est une permutation et conserve donc le poids des mots), donc on peut décoder cette erreur et retrouver le message initial m S. Il ne reste plus qu'à multiplier par S^{-1} pour retrouver le message m et avoir fini de déchiffrer.

4.4 Exemple de Chiffrement

Soit le code de Hamming (7, 4) avec G matrice génératrice qui corrige toutes les erreurs simples. On choisit la matrice de brouilleur S et une matrice de permutation P. On calcul la clé publique qui correspond à la matrice G'.

$$G = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix} \quad P = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{vmatrix}$$

$$S = \begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{vmatrix} \quad G' = S\,G\,P = \begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{vmatrix}$$

Soit X = (1 1 0 1) le message à envoyer. Si on suppose que le canal de transmission introduit une erreur simple de poids 1 de valeur e = (0 0 0 0 1 0 0). Au lieu d'envoyer le message X c'est un autre message Y qui est envoyé.

$$y = xG' + e = (0110010) + (0000100) = (0110110) \quad (10)$$

A la réception de Y' on calcul d'abord : $y' = yP^{-1}$

$$P^{-1} = \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{vmatrix} \quad S^{-1} = \begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{vmatrix}$$

$$yP^{-1} = (xG' + e)P^{-1} = xSG + eP^{-1} = xSG + e' \quad (11)$$

e' est un vecteur du poids t (puisque P^{-1} est également une matrice de permutation). On obtient : y '= (1 0 0 0 1 1 1)

On applique le décodage de Meggit pour déterminer le vecteur d'erreur e' et par conséquent le mot de code (xS)G. Le syndrome de y' trouve est $(1\ 1\ 1\ 0)^T$, ainsi l'erreur se produit en position 7 (détails omis). Le récepteur a maintenant le mot de code y = (1 0 0 0 1 1 0).

Le vecteur xS= (1 0 0 0) peut maintenant être obtenu en multipliant par G^{-1} du côté droit (cependant, on peut écrire G sous le format standard [I_k A], et alors les xS seraient juste les premières positions de k du xSG et cette multiplication ne serait pas nécessaire). En conclusion, on obtient x = (1 0 0 0) S^{-1} = (1 1 0 1) en multipliant des xS du côté droit par S^{-1}.

5. Conclusion

Nous avons introduit dans cette communication une nouvelle application de la théorie des codes correcteurs d'erreurs en cryptographie. Cette application est bien concrète, mais elle nécessite certainement encore quelques petits ajustements avant d'être réellement utilisable en pratique. Car on a pu recensé trois inconvénients pour ce crypto système de McEliece.

1. La taille de la clef publique (G') est grande. Ceci posera certainement des problèmes d'exécution.

2. Le message chiffré est plus long que le message clair. Cette augmentation de la largeur du message chiffré rend le système plus sensible aux erreurs de transmission.

3. Le crypto système n'est employé pour la signature ou l'authentification parce que l'algorithme de chiffrage n'est pas linéaire et tout l'algorithme est vraiment asymétrique.

La sécurité d'un système s'évalue grâce au coût des meilleures attaques, mais l'effort fourni pour essayer d'en trouver de meilleures est aussi une part importante de cette sécurité. De ce point de vue là, un système ancien et bien connu de tous est souvent préférable à un système plus jeune, même s'il offre de bonnes propriétés.

6. Bibliographies

[1] G.C Clark, J.B Cain ,'' Error Correcting Coding for Digital Communication'', Plenum Press 1981.

[2] D.J Costello, S.Lin,'' Error Control Coding : Fondamentals and Applications'', Prentice Hall 1983.

[3] AL. Spataru, 'Transmission de l'Information II : Codes et Décisions Statiques', MASSON et CIE, 1973.

[4] A. Poli, Li Huguet, '' Codes Correcteurs : Théorie et Applications'', Masson Paris 1989.

[5] S.Foughali, S.Khelifa, '' Concaténation des Codes Cyclique (Reed Solomon – Hamming) Appliquées aux images fixes'', Institut d'Informatique, USTO 1998.

[6] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Prog. Rep.*, Jet Prop. Lab., California Inst. Technol., Pasadena, CA, pages 114–116, January 1978.

[7] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[8] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.