

Méthodes de Déterminations des Polynômes Primitifs

N. HADJ-SAID*, A. ALI-PACHA*, A. M'HAMED** et A. BELGHORAF*

**Université des Sciences et de la Technologie d'Oran BP 1505 El M'Naouer Oran 31036 ALGERIE*

nim_hadj@yahoo.fr

belghoraf_daman@yahoo.fr

alipacha@yahoo.com

***Institut National des Télécommunications, Evry France*

abdallah.mhamed@it-sudparis.eu

Résumé: La théorie derrière la génération de bits pseudo aléatoires a beaucoup en commun avec celle qui gouverne le fonctionnement de codes polynomiaux CRC, détection et correction des erreurs (codage de l'information), et les techniques utilisées pour les mettre en œuvre par programmation sont très semblables. Ils utilisent des opérations sur des symboles d'information appartenant à un corps fini GF(2).

La connaissance des structures algébriques ainsi que les propriétés des polynômes primitifs permettent d'établir la construction pratique des codes détecteurs et correcteurs d'erreurs et, des générateurs pseudos aléatoires.

Notre communication consiste à proposer une méthode de recherche des polynômes primitifs de très grands poids et de degré supérieur, la méthode ainsi proposé est implémenté sous la version 7.0 du MATLAB.. La connaissance de ces polynômes primitifs est une étape algorithmique importante pour les ingénieurs concepteurs des applications des télécommunications.

Mots clés : Codage, Chiffrement Continu, Pseudo Aléatoire, Polynôme, Polynôme Primitif, Idéal.

1. INTRODUCTION

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'information 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.

La théorie derrière la génération de bits pseudo aléatoires a beaucoup en commun avec celle qui gouverne le fonctionnement de codes polynomiaux CRC, détection et correction des erreurs (codage de l'information), et les techniques utilisées pour les mettre en œuvre par programmation sont très semblables. Ils utilisent des opérations sur des symboles d'information appartenant à un corps fini GF(2).

La connaissance des structures algébriques ainsi que les propriétés des polynômes primitifs permettent d'établir la construction pratique des codes détecteurs

et correcteurs d'erreurs et, des générateurs pseudos aléatoires. Les applications que nous avons étudiées :

- 1 Codage de l'information,
- 2 Sécurité des données et
- 3 Génération des bruits de simulations

Sont engendres par un concept mathématique particulier qui est le polynôme primitif.

Notre communication consiste à proposer une méthode de recherche des polynômes primitifs de très grands poids et de degré supérieur.

La connaissance de ces polynômes primitifs est une étape algorithmique importante pour les ingénieurs concepteurs des applications de télécommunications.

2. L'ANNEAU DES POLYNOMES

On appelle polynôme une expression de la forme :

$$f(x)=a_0+a_1x+a_2x^2+\dots+a_nx^n$$

Dans laquelle les coefficients a_0, a_1, \dots, a_n appartient à un corps de Galois GF(2).

On vérifie facilement que les polynômes ont une structure d'anneau de variable x.

Le polynôme est dit normaliser si le coefficient du terme de plus haut degré est 1.

Le degré d'un polynôme est la puissance la plus élevée de x à coefficient non nul.

2.1 POLYNOME IRREDUCTIBLE :

Soit $GF(2)[x]$ l'ensemble des polynômes en x [10] à coefficient dans $GF(2)$, un polynôme $g(x)$ de $GF(2)[x]$ est dit irréductible sur $GF(p)$, s'il ne se décompose pas en un produit de polynôme non-triviaux, c'est à dire polynômes de degré strictement positifs de $GF(p)[x]$.

- ✓ Le polynôme $P(x) = 1+x+x^2$ est irréductible sur $GF(2)$.
- ✓ Le polynôme $F(x) = x^3 + x^2 + x + 1$ n'est pas irréductible sur $GF(2)$ car $F(x) = (x+1)(x^2+x+1)$.

2.2 PERIODE D'UN POLYNOME

Tout polynôme à une période, est la période d'un polynôme irréductible de degré n est $2^m - 1$

Tous polynômes irréductibles sur $GF(2)$ de degré m divise $x^l + 1$ avec $l = 2^m - 1$. [4, 5]

- ❖ $x^3 + x + 1$ divise $x^7 + 1$ on effet $2^3 - 1 = 7$
- ❖ $x^7 + 1 = (x^4 + x^2 + x + 1)(x^3 + x + 1)$

2.3 POLYNOME PRIMITIF :

Un polynôme $p(x)$ de degré m est dit primitif [1] si le plus petit entier n pour que :

$$g(x) \text{ divise } x^n + 1 \text{ est } n = 2^m - 1.$$

2.4 IDEAL SUR UN ANNEAU DE POLYNOME:

Un ensemble de polynômes sur $GF(2)$ est un idéal si et seulement s'il contient tous les multiples d'un polynôme donné $g(x)$.

3. DETERMINATION DES POLYNOMES PRIMITIFS

Ces polynômes générateurs sont alors déterminés en factorisant le polynôme $x^n + 1$ en un produit de polynômes irréductibles qui ont leur période égale au le plus petit entier u tel que ce dit polynôme divise $x^u + 1$. Afin de trouver ces polynômes, nous allons implanter notre logiciel sous Matlab version 7.0., dans un ordinateur doté d'un micro processeur Pentium III et d'une vitesse de 1 GHz avec un disque dur de capacité de 20 Go et une RAM de 128 Mo. Notre méthode suit trois étapes [9] :

3.1 RECHERCHE DES POLYNOMES DIVISEURS : 1^{ERE} ETAPE

Dans cette étape on cherche tous les polynômes diviseurs de $x^n + 1$ sans aucune restriction (i.e. triviaux ou non) sur ces diviseurs (voir figure N°1), puis en les mémorisant dans une case mémoire A.

3.2 RECHERCHE DES POLYNOMES IRREDUCTIBLES : 2^{EME} ETAPE

Cette deuxième étape on travail sur la case mémoire A qui contient tous les diviseurs de $x^n + 1$, on cherche parmi ces diviseur ceux qui ont irréductibles. Ces polynômes irréductibles diviseurs de $x^n + 1$ sont mémorisés dans une case mémoire B (Voir figure N°2).

Un polynôme $g(x)$ est dit irréductible s'il ne possède aucuns diviseurs de degré supérieur à zéro.

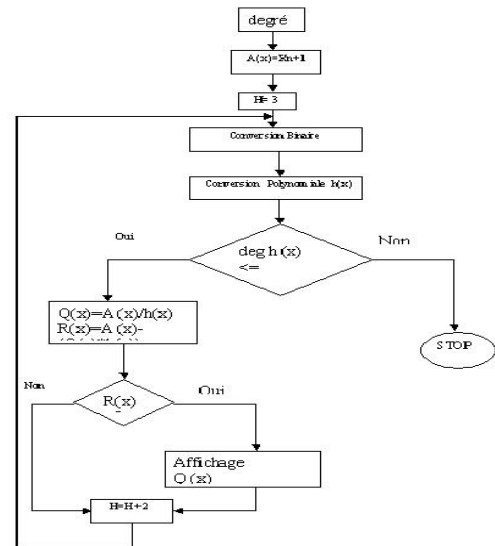


Figure 1 : recherche des polynômes diviseurs

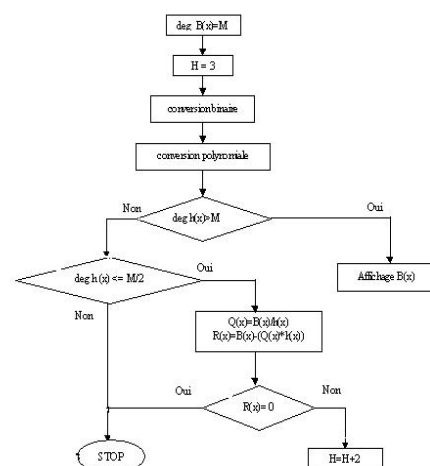


Figure 2 : recherche des polynômes irréductibles

3.3 RECHERCHE DES POLYNÔMES PRIMITIFS : 3^{ÈME} ÉTAPE

Dans troisième étape on travail sur la case mémoire B en choisissant parmi ces polynômes ceux qui ont la caractéristique d’avoir la même période ou ordre du polynôme. Cette période est le plus petit entier u tel que ce dit polynôme divise x^u+1 .

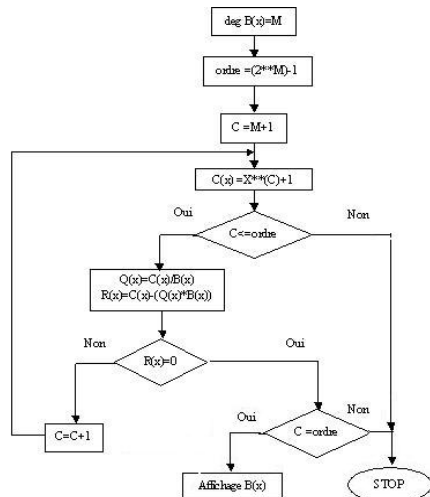


Figure 3 : recherche des polynômes primitifs

- Si un polynôme $g(x)$ de degré r est irréductible, on peut montrer que la période de $g(x)$ divise 2^r-1 .
- Lorsque la période de $g(x)$ est égale à 2^r-1 , le polynôme irréductible $g(x)$ est appelé primitif.

Théorème *Un polynôme primitif est irréductible et satisfait les deux conditions suivantes :*

- 1- il a un nombre impair de termes (le terme 1 inclus)
- 2- si il est de degré supérieur à 3 alors $P(x)$ doit se diviser en $(1+x^k)$, avec $k=2^n-1$

4. RESULTATS ET INTERPRETATIONS :

Nous avons procéder à la recherche de ces polynômes générateurs qui sont déterminés en factorisant le polynôme x^n+1 [9] en un produit de polynômes irréductibles, le tableau ci-dessous nous donne quelques résultats obtenus.

D’après ce tableau on peut déduire :

1. Tous les polynômes générateurs sont issus de la division de x^n+1 , avec $n=2^k-1$ (k entier).
2. Soit $n_1=2^k-1$ (k entier), $\forall m$ un entier, tel que $n=m*n_1$, les polynômes générateurs issus de la division x^n+1 sont les mêmes que ceux issus de la division $x^{n_1}+1$.
3. Si n n’est pas un multiple de 2^k-1 (k entier), Alors tous les polynômes diviseurs de x^n+1 ne sont pas des

polynômes générateurs.

Enfin, on peut constater que pour la recherche des polynômes générateurs pour un n donné, il faut :

- Décomposer n en un produit de facteurs de la forme 2^k-1 (k un entier).
- Les polynômes générateurs issus de x^n+1 sont la somme des polynômes générateurs issus des différents facteurs.

DEGRE	POLYNOMES GENERATEURS
2 et 3	$x^2 + x + 1 : x^3 + x + 1 : x^3 + x^2 + 1$
4 et 5	$x^4 + x + 1 : x^4 + x^3 + 1 : x^5 + x^2 + 1 : x^5 + x^3 + 1 : x^5 + x^3 + x^2 + x + 1 : x^5 + x^4 + x^3 + x + 1 : x^5 + x^4 + x^3 + x^2 + 1$
6	$x^6 + x^5 + x^2 + x + 1 : x^6 + x^5 + x^3 + x^2 + 1 : x^6 + x^5 + x^4 + x + 1 : x^6 + x + 1 : x^6 + x^4 + x^3 + x + 1$
7	$x^7 + x^4 + x^3 + x^2 + 1 : x^7 + x^4 + 1 : x^7 + x^5 + x^2 + x + 1 : x^7 + x^3 + x^2 + x + 1 : x^7 + x^5 + x^4 + x^3 + 1 : x^7 + x + 1 : x^7 + x^3 + 1 : x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$
8	$x^8 + x^4 + x^3 + x^2 + 1 : x^8 + x^5 + x^4 + x^3 + 1 : x^8 + x^5 + x^4 + x^3 + x^2 + x + 1 : x^8 + x^6 + x + x^2 + 1 : x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$
9	$x^9 + x^4 + 1 : x^9 + x^4 + x^3 + x + 1 : x^9 + x^5 + 1 : x^9 + x^5 + x^3 + x^2 + 1 : x^9 + x^5 + x^4 + x + 1$
10	$x^{10} + x^3 + 1 : x^{10} + x^3 + x^2 + x^1 + 1 : x^{10} + x^4 + x^3 + x^1 + 1 ; x^{10} + x^4 + x^3 + x^2 + 1$
11	$x^{11} + x^2 + 1 : x^{11} + x^4 + x^2 + x^1 + 1 : x^{11} + x^5 + x^3 + x^1 + 1 : x^{11} + x^6 + x^2 + x^1 + 1 : x^{11} + x^6 + x^5 + x^1 + 1 : x^{11} + x^6 + x^5 + x^2 + 1$
12	$x^{12} + x^3 + 1 : x^{12} + x^3 + x^2 + x^1 + 1 : x^{12} + x^4 + x^2 + x^1 + 1 : x^{12} + x^5 + 1 : x^{12} + x^5 + x^4 + x^1 + 1 : x^{12} + x^5 + x^4 + x^2 + 1$
13	$x^{13} + x^4 + x^3 + x^1 + 1 : x^{13} + x^5 + x^2 + x^1 + 1 : x^{13} + x^5 + x^4 + x^2 + 1 : x^{13} + x^6 + x^4 + x^1 + 1 : x^{13} + x^6 + x^5 + x^2 + 1 : x^{13} + x^6 + x^5 + x^3 + x^2 + x^1 + 1$

4.1 EXEMPLE :

Pour $n=1023 = 3*11*31$. Donc, les polynômes générateurs de $x^{1023}+1$ sont la somme des polynômes générateurs de 3 et 31 plus les générateurs de degré 10 :

$$x^2 + x + 1, x^5 + x^2 + 1, x^5 + x^3 + 1, x^5 + x^3 + x^2 + x + 1, x^5 + x^4 + x^2 + x + 1, x^5 + x^4 + x^3 + x + 1, x^5 + x^4 + x^3 + x^2 + 1$$

4.2 INCONVENIENTS DE LA METHODE

Nous avons noté que notre logiciel donne des résultats dans un temps de calcul réduit pour la recherche des polynômes de degré inférieur à 8, par contre il prend beaucoup de temps de calcul lorsqu'on augmente le degré du polynôme recherché par exemple pour le degré 13 nous avons attendu plus de 4 jours (100 heures) de calcul pour trouver ces résultats.

5. CONCLUSION

Les polynômes primitifs de très grands poids et de degré supérieur sont la base de ces deux importantes familles d'applications en télécommunications :

- Détection et correction d'erreurs CRC : techniques utilisées en transmission de données pour assurer la fiabilité et l'intégrité des transmissions.
- Génération pseudo aléatoire : les techniques de génération de bits pseudo aléatoires qui sont au centre de nombreuses applications importantes en télécommunications: simulation, test, brouillage, chiffrement et étalement spectral. Dont les fondements théoriques sont apparentés.

La connaissance parfaite de ce concept et sa détermination en particulier sont un outil de base pour les ingénieurs concepteurs des applications de télécommunications.

BIBLIOGRAPHIES

- [1] Kada ALLAB, "élément d'analyse", Entreprise Nationale du livre , OPU Alger 1990.
- [2] Bruce Schneier, "Applied Cryptography, Protocols, Algorithms, and source Code in C" , edition John Wiley & Sons Inc., 1994.
- [3] Gilles Zémor, «Cours de Cryptographie», CASSINI, 2000.
- [4] George CULLMANN, code correcteur et détecteur d'erreurs, Paris 1967.
- [5] George CULLMANN, Codage et transmission de l'information, Paris 1968.
- [6] C. DUFAZA, G. CAMBON, "LFSR based deterministic and pseudo-random test pattern generator structures", Proc. ETC'91, pp. 27-34, 1991.
- [7] P.H. Bardell, "Analysis of Cellular Automata Used as Pseudorandom Pattern Generators", Proc. of International Test Conference, pp. 762-767, 1990.
- [8] [www.mea.isim.univ-montp2.fr/POLYCOPS/EST_\(C.LANDRAULT\)/CHAP8.pdf](http://www.mea.isim.univ-montp2.fr/POLYCOPS/EST_(C.LANDRAULT)/CHAP8.pdf)
- [9] A.S.SLAIMI et A.BOUMAIZA, "Recherche des Polynômes Générateurs et leur Application dans les Télécommunications", Mémoire d'ingénieur en d'Informatique USTOran 2003.
- [10] A.POLI et L.Hugest , Codes correcteurs :théorie et applications, (Paris , Masson, 1989).
- [11] M.DJELLOULI et N.SADOUKI, "Etude et Implémentation d'une Méthode de Chiffrement Continue", Mémoire d'ingénieur en d'Electronique USTOran 2003.
- [12] S.Foughali, S.Khelifa, " Concaténation des Codes Cyclique (Reed Solomon – Hamming) Appliquées aux images fixes", Institut d'Informatique, USTO 1998.