

Polynômes irréductibles dans $\mathbb{F}_2[X]$, codes correcteurs BCH et QR codes

Kevin Trancho
L3 Informatique

Université Paris-Est Marne-la-Vallée
Projet tuteuré encadré par Marc ZIPSTEIN et Jean-Yves THIBON

20 Juin 2017

Sommaire

- 1 Introduction
- 2 Recherche de Polynômes irréductibles dans $\mathbb{F}_2[X]$
 - Crible d'Eratosthène au degré n
 - Recherche par factorisation récursive
- 3 codes BCH
 - Construction
 - Codage et Correction d'erreurs
- 4 QR codes
 - Construction données
 - Ecriture dans le QR code et correction des erreurs
- 5 Démonstration
- 6 Conclusion

Sujet

- Comprendre notion de Corps fini et recherche de polynômes irréductibles $\mathbb{F}_2[X]$
- Initiation codes correcteurs BCH et codes Reed-Solomon
- Création d'une version du QR code
- Codé en 10 000 lignes de C pour plus de 400 fonctions dans 31 fichiers sources

Implémentations de base

- $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$
- Addition et soustraction implémentées par le 'xor'
- Multiplication décalage bit-à-bit selon degré des monômes et addition par 'xor'
- Recherche de polynômes irréductibles pour calcul d'inverse et construction de corps finis $\mathbb{F}_{2^{deg(p)}} = \mathbb{F}_2[X]/p(X)$
- Calcul des inverse par implémentation identité de Bezout :

$$PGCD(p(x), q(x)) = u(x)p(x) + v(x)q(x)$$

$$r_k(x) = u_k(x)p(x) + v_k(x)q(x)$$

Crible d'Eratosthène au degré n

- Utilise des polynômes codés sur un entier pour plus de vitesse (limité au degré 31)
- Fabriquer liste des polynômes de degré n
- Calculer le produit de tous les polynômes irréductibles p_1, p_2, \dots, p_q de degré inférieur à n
- Estimer irréductible tout polynôme non éliminé par le calcul de produits



Recherche par factorisation récursive

- Générer aléatoirement un polynôme de degré n quelconque
- Tester l'irréductibilité :
 - Tester s'il divise $X^{2^n} - X$
 - Factoriser par l'algorithme de Berlekamp
- Si non irréductible, appliquer le test récursivement jusqu'à trouver tous les facteurs irréductibles
- Plus haut polynôme irréductible trouvé de degré 11 457

Construction BCH

- BCH taille n pouvant corriger t erreurs
- Choisir un polynôme primitif et un élément primitif α
- Construire g le polynôme minimal tel que

$$g(\alpha) = g(\alpha^3) = \dots = g(\alpha^{2^t-1})$$

- Calcul des $M_i \in \mathbb{F}_2[X]$ tels que

$$M_i(x) = (x - \alpha^i)(x - \alpha^{2i}) \dots (x - \alpha^{2^{k-1}i}), \quad k > 0, \quad 2^k i \equiv i[n]$$
- Résolution système par implémentation méthode de Cramer
- $g(x) = PPCM(M_1(x), M_3(x), \dots, M_{2^t-1}(x))$

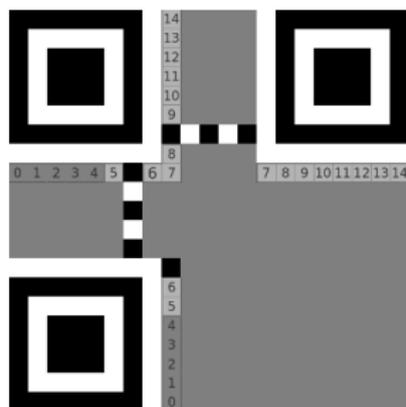
Codage et Correction d'erreurs

- Calcul du reste du message m modulo g
- Calcul des syndromes, évaluation des $s_i = m(\alpha^i)$, $i \in \llbracket 1, 2t \rrbracket$
- Calcul du polynôme localisateur σ avec $\deg(w) < t$:

$$\sigma(x)s(x) + v(x)x^{2t} = w(x)$$

- Chien search pour correction des erreurs

Construction format



- Format codé par BCH : taux de correction d'erreurs et masque
- 'xor' message codé et '101010000010010'

Construction données

- Mode : Numérique, Alphanumérique et Bytes implémentés
- Codes de Reed-Solomon pour correction de t_c erreurs, message taille t_m , taille totale des données à écrire t
- Construction de $\mathbb{F}_{256} = \mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X^2 + 1)$
- $$g(x) = \prod_{i=0}^{t-t_m-1} (x - \alpha^i)$$
- Décomposition en blocs [Données - Correction Erreurs]

Correction des erreurs

- Calcul des syndromes, polynôme localisateur et chien search comme pour les BCH
- Calcul de l'erreur par algorithme de Forney :
 - $\Omega(x) = s(x)\sigma(x) \pmod{x^{t_c}}$
 - valeur de l'erreur en position i :

$$\frac{\alpha^i \Omega(\alpha^{-i})}{\sigma'(\alpha^{-i})}$$

Démonstration



- Création d'un QR code
- Décodage d'un QR code avec erreurs

Conclusion

- Initiation à la cryptographie intéressante et très instructive en continuité des notions vues en Mathématiques pour l'Informatique 4
- Intérêt pour une utilisation contemporaine : QR codes
- Merci de votre attention
- Posez vos questions