

TD2 - LFSR et Chiffrement par Flot

Responsable : M. Bombar

1 Introduction aux LFSR

- (Q1) Écrivez une fonction `LFSR_step(P, state)` qui simule l'étape d'un LFSR. Elle prend en entrée un polynôme de rétroaction $P \in \mathbb{F}_q[X]$ ainsi que l'état S_t au temps t , et doit ressortir l'état au temps $t + 1$, ainsi que la sortie à l'instant t . On ne suppose pas que le corps de base est \mathbb{F}_2 .
- (Q2) Écrivez une fonction `LFSR(P, state, N)` qui prend un polynôme de rétroaction, l'état initial S_0 ainsi qu'un entier N , et retourne les N premiers éléments de sortie du LFSR de rétroaction P , initialisé par S_0 .
- (Q3) Vérifiez que le LFSR de longueur 5 et de polynôme de rétroaction $P(X) \stackrel{\text{def}}{=} 1 + X^4 + X^5 \in \mathbb{F}_2[X]$, initialisé par $S_0 \stackrel{\text{def}}{=} (0, 0, 1, 0, 1)$ retourne une suite débutant par $0, 0, 1, 0, 1, 0, 1, \dots$. Quelle est sa période ?
- (Q4) Écrivez une fonction qui vérifie si la suite produite par un LFSR satisfait aux critères de Golomb, et tester avec plusieurs polynômes de rétroactions, et états initiaux.

On peut former l'anneau des séries formelles en une indéterminée avec Sage en suivant l'exemple suivant :

```
sage: F2 = GF(2)
sage: PS.<X> = PowerSeriesRing(F2)
```

- (Q5) Écrivez une fonction qui prend en entrée un LFSR et retourne son polynôme de rétroaction minimal.

2 Cycles des registres d'un LFSR

- (Q6) Écrivez une fonction prenant en paramètre l'état initial S_0 d'un LFSR, et son polynôme de rétroaction P , et retourne l'ensemble des différents états internes S_0, S_1, \dots, S_{T-1} obtenus sur une période.
- (Q7) Testez avec $P(X) = 1 + X^4 + X^5 \in \mathbb{F}_2[X]$ et les états initiaux $(0, 0, 1, 0, 1), (0, 1, 0, 0, 0), (1, 0, 1, 0, 0)$. Que remarquez-vous ?
- (Q8) Écrivez une fonction prenant en paramètre le polynôme de rétroaction et retourne l'ensemble de tous les cycles possibles (en faisant varier l'état initial).
- (Q9) Combien de cycles sont produits par le LFSR de longueur 5 et de polynôme de rétroaction P ? Même question avec les LFSR de longueur 4 de polynômes de rétroaction binaires
- $1 + X^2 + X^4$
 - $1 + X + X^2 + X^3 + X^4$
 - $1 + X + X^4$
- (Q10) Testez dans les différents cas si les polynômes précédents sont minimaux.

3 Algorithme de Berlekamp-Massey

Les LFSR semblent de prime abord fournir une solution clé en main pour générer des chaînes de bits pseudoaléatoires :

- Ils sont extrêmement rapides, surtout en hardware.
- Il est très facile de produire des suites aux périodes arbitrairement longues.
- Les suites maximales vérifient les critères de Golomb.

En revanche, le but de cet exercice est de voir que ceci ne suffit pas pour fournir des chiffrements par flot sécurisés.

Soit $\mathbf{s} \stackrel{\text{def}}{=} (s_n)_{n \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$ une suite ultimement périodique. On note $\Lambda(\mathbf{s}) \geq 1$ sa complexité linéaire.

(Q12) Montrez que $2\Lambda(\mathbf{s})$ termes consécutifs de la suite caractérisent entièrement \mathbf{s} . Plus précisément, montrez que si on connaît $\Lambda(s)$, ainsi que $2\Lambda(s)$ termes *consécutifs* de \mathbf{s} , alors il est possible de calculer son polynôme de rétroaction minimal P .

Indication : Écrire les coefficients de P comme les inconnues d'un système linéaire.

On suppose que l'on connaît une borne supérieure $\Lambda(s) \leq \ell$ sur la complexité linéaire de \mathbf{s} . Déterminez un algorithme qui prend en entrée \mathbf{s} et retourne $\Lambda(\mathbf{s})$ ainsi que le polynôme de rétroaction minimal en temps $O(\ell^4)$. Quelle est sa complexité en mémoire ?

La Question (Q12) nous donne déjà un algorithme *polynomial* pour caractériser la suite \mathbf{s} en n'observant qu'un nombre restreint de coefficients. En particulier, même si \mathbf{s} a d'excellentes propriétés statistiques, elle n'est absolument pas *imprédictible*. Cependant, la complexité $O(\ell^4)$ peut parfois être assez grosse pour monter des attaques en pratique, surtout lorsque les LFRS sont combinés à d'autres opérations, ou lorsque la borne supérieure ℓ n'est pas connue. Cependant, on va voir qu'il est possible de faire beaucoup mieux étant donnée la *structure* des systèmes linéaires mis en jeu.

Pour cela, on va décrire, et implémenter, un algorithme dû à James Massey [Mas69], qui a montré comment une procédure proposée deux ans plus tôt par Elwyn Berlekamp [Ber68] afin de décoder une certaine famille de codes correcteurs d'erreurs appelés codes BCH, pouvait être utilisée pour déterminer le LFSR minimal qui génère une suite donnée. Plus précisément, étant donnée une suite $\mathbf{s} = (s_0, \dots, s_{n-1})$ de longueur n , ou plus précisément la suite prolongée par périodicité, l'algorithme de Berlekamp-Massey va faire n itérations. À l'itération t , l'algorithme de Berlekamp-Massey détermine un LFSR de longueur minimale qui produit les t premiers symboles de \mathbf{s} .

Algorithm 1: Algorithme de Berlekamp-Massey

Input: $\mathbf{s}^{(n)} = (s_0, \dots, s_{n-1})$ une suite de n éléments de \mathbb{F}_q .

Output: $\Lambda(\mathbf{s}^{(n)})$ la complexité linéaire de $\mathbf{s}^{(n)}$ et P , le polynôme de rétroaction de degré minimal qui engendre $\mathbf{s}^{(n)}$.

```

1 /* Initialisation */
2  $P(X) \leftarrow 1$ 
3  $Q(X) \leftarrow 1$ 
4  $\Lambda \leftarrow 0$ 
5  $m \leftarrow -1$ 
6  $d' \leftarrow 1$ 
7 /* Algorithme */
8 for  $t \in \{0, \dots, n-1\}$  do
9    $d \leftarrow s_t + \sum_{i=1}^{\Lambda} p_i s_{t-i}$ 
10  if  $d \neq 0$  then
11     $T(X) \leftarrow P(X)$  ▷ Polynôme temporaire
12     $P(X) \leftarrow P(X) - d(d')^{-1}Q(X)X^{t-m}$ 
13    if  $2\Lambda \leq t$  then
14       $\Lambda \leftarrow t + 1 - \Lambda$ 
15       $m \leftarrow t$ 
16       $Q(X) \leftarrow T(X)$ 
17       $d' \leftarrow d$ 
18 return  $\Lambda$  et  $P(X)$ 

```

(Q13) Décrire les étapes de l'algorithme de Berlekamp-Massey appliqué à la suite binaire de longueur 7 et définie par

$$\mathbf{s}^{(7)} \stackrel{\text{def}}{=} (0, 1, 1, 1, 1, 0, 0).$$

(Q14) Implémentez l'algorithme de Berlekamp-Massey avec Sage.

(Q15) Quelle est sa complexité?

(Q16) Montrez que l'algorithme est correct, c'est-à-dire qu'à la fin de l'algorithme on a bien $\Lambda = \Lambda(\mathbf{s}^{(n)})$ et que P est bien le polynôme de rétroaction minimal de $\mathbf{s}^{(n)}$.

4 Challenge

Références

- [Ber68] Elwyn R. BERLEKAMP. “Factoring Polynomials over finite fields”. In : *Algebraic Coding Theory*. Sous la dir. d’E. R. BERLEKAMP. McGraw-Hill, 1968. Chap. 6.
- [Mas69] James L. MASSEY. “Shift-Register Synthesis and BCH Decoding”. In : *IEEE Trans. Inform. Theory* 15.1 (1969), p. 122-127.