

# Cryptanalyse

## Cours 2 - Chiffrement par Flot (Stream Ciphers)

Maxime Bombar

08 Septembre 2026

# Introduction

# Rappel de la semaine dernière : One-Time-Pad

- Chiffrement sûr au sens de la théorie de l'information (*Perfect secrecy*)
- Clés nécessitent une grosse entropie :  
e.g. clés aléatoires et aussi longues que le message
- Pas adaptés pour tous les contextes.

Mais les adversaires ont des ressources *bornées*.

→ Chiffrement par flot : Clé **pseudo-aléatoire**.

# Rappels sur les chiffrements par flots

Un **chiffrement par flots** (additif et synchrone) est obtenu en ajoutant au message, une suite chiffrante **pseudoaléatoire** (appelée aussi *keystream*) générée indépendamment :

$$\forall k, \quad c_k = m_k + s_k$$

La suite chiffrante ( $s_k$ ) est obtenue à partir d'une graine courte, et d'un **générateur pseudo-aléatoire cryptographique** (CPRG).

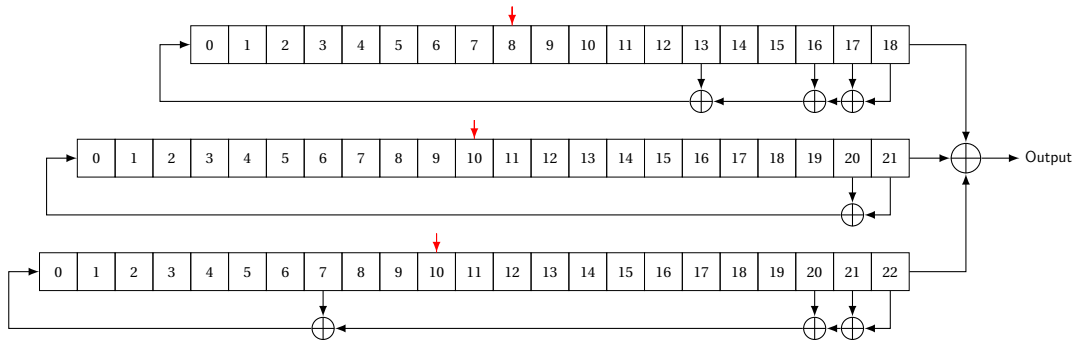
# Avantages des Chiffrements par Flot

- PRG implantés en **hardware**.
- En général, extrêmement rapides.
- Peu gourmands en énergie.

→ Utilisés sur des terminaux aux ressources limitées.

- WEP (Wifi, norme IEEE 802.11, 1999) : Chiffrement RC4.
- Bluetooth : Chiffrement E0.
- 2G/3G : Chiffrement A5/1.
- 4G/5G : Chiffrement SNOW.
- Communications radio par l'armée (utilisation historique).

# Exemple : Chiffrement par Flot A5/1 (Cassé)



Attaques extrêmement efficaces (quelques secondes d'écoute du trafic, quelques minutes de temps de calcul).

# Déclin du Chiffrement par Flot (1970's – 2000's)

- Télécommunications : Analogique → Découpage par paquets.
  - DES (Data Encryption Standard), standardisé en 1976.
  - Historiquement, sécurité par l'obscurantisme (contraire à Kirchhoff!).
  - Cryptanalyse spécifique des chiffrements par flots.
- 
- RC4 rétroingéniéré en 1994, spécifications sur une *mailing list* : Cipherpunks.
  - WEP utilise RC4, mais en pire → Surnommé Weak Encryption Protocol (cryptanalyse complète dès 2001).
  - <https://github.com/aircrack-ng/aircrack-ng/>



# Stream Ciphers: Dead or Alive?

---

Adi Shamir  
Computer Science Dept  
The Weizmann Institute  
Israel

ASIACRYPT 2004

# Regain d'attractivité ?

- Compétition eSTREAM (2004-2008) → 7 chiffrements, dont Salsa20.
- Renaissance des besoins « **ressources limités** » et « **sobriété** »

- ChaCha20 : Variante de Salsa20 avec de meilleures performances.
- Fait partie de TLS1.3. Utilisé notamment dans **DNS over TLS**.

kdig +tls u-bordeaux.fr @80.67.169.12 // DNS de la FDN, FAI Associatif.

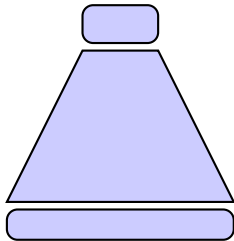
# Choix du PRG

La sécurité d'un chiffrement par flots est directement reliée au caractère pseudo-aléatoire du PRG choisi.

Attention au choix du PRG ! (cf soutenances de la semaine dernière).

# PseudoRandom (Number) Generators

# Caractéristiques d'un PRG



- Algorithme **déterministe** pour générer de l'aléa.
  - Sortie entièrement déterminée par la graine initiale.
  - Sortie doit **ressembler** à une chaîne uniformément aléatoire.
  - Bonus : Rapide.
- 
- Bonnes propriétés statistiques → **pas suffisant**
  - Nécessite **Non predictabilité** :

$(s_n, \dots, s_{n+m})$  ne doit pas permettre de retrouver  $s_{n-1}$  ou  $s_{n+m+1}$  efficacement.

# Une remarque importante

Soit  $\ell$  la taille de l'état interne. Alors le PRG peut prendre **au plus**  $2^\ell$  états internes différents, et la suite  $(s_t)$  est **ultimement périodique** de période au plus  $2^\ell$ .

On va chercher à produire des suites chiffrantes de **période maximale**.

# Vecteur d'Initialisation (IV)

Réutiliser la même clé secrète résulte en **la même suite chiffrante**.  
→ même problème que le *One-Time Pad*.

- L'état interne initial  $S_0$  est déterminé à partir de la clé privée  $K$  et d'une entrée auxiliaire **publique** appelée **vecteur d'initialisation** (IV).
- L'IV est changé régulièrement et permet de réutiliser la même clé  $K$ .

# Éléments constitutifs d'un PRG

- **Fonction d'initialisation** : Détermine l'état initial  $S_0$  à partir de la clé privée  $K$  et de l'IV.
- **Fonction de transition** : Modifie la valeur de l'état interne :  $S_{t+1} = \Phi(S_t)$  à chaque tic d'horloge. En d'autres termes,  $S_t = \Phi^t(S_0)$  (avec  $\Phi^{t+1} \stackrel{\text{def}}{=} \Phi \circ \Phi^t$ ).
- **Fonction de filtrage** : Retourne un symbole  $s_t$  en fonction de l'état interne  $S_t$ .

Toutes ces fonctions dépendent éventuellement de  $K$  et de l'IV.

# Modèles d'attaque

- **Key-Recovery** : On cherche à retrouver la clé secrète à partir de quelques termes de la suite produite par le PRG.
- **État Initial** : On cherche à récupérer l'état initial. Remarque : récupérer l'état à un instant  $t$  est souvent suffisant.
- **Prédiction** : À partir de termes consécutifs de la suite chiffrante, on cherche à prédire le(s) bit(s) suivant(s).
- **Distingueur** : À partir de  $N$  termes, on cherche à savoir s'ils ont été produits par un PRG ou sont des bits uniformément aléatoires.

# Attaques Génériques

## Pour aller plus loin

Les **fonctions aléatoires** ont des propriétés **très particulières** qu'il est potentiellement possible d'exploiter pour analyser une suite produite par un PRG.



Philippe Flajolet, Robert Sedgewick - *Analytic Combinatorics* - 2009.

## Compromis Temps-Mémoire

Il existe des attaques par compromis temps-mémoire spécifiquement dédiées aux chiffrements par flots (e.g. Hellman 1980, ou Biryukov, Shamir 2000).

$$\text{Temps} \times \text{Mémoire}^2 \times \text{Données}^2 = |\text{Espace}|^2$$

# Critères Statistiques

# Runs

On considère une suite  $(s_n)_{n \in \mathbb{N}}$  binaire, périodique de période  $T$ .

## Définition

On appelle *run* une succession consécutive de bits identique maximale.

## Exemple :

Dans la suite prenant les valeurs (0100011), l'ensemble des *runs* est :

- 0 : Un run de taille 1,
- 1 : Un run de taille 1,
- 000 : Un run de taille 3,
- 11 : Un run de taille 2.

# Autocorrélation

On considère une suite  $(s_n)_{n \in \mathbb{N}}$  binaire, périodique de période  $T$ .

## Définition

La fonction d'autocorrélation de  $z$  est

$$C_s(\tau) \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} (-1)^{s_i + s_{i+\tau}}.$$

**Remarque :**

$$C_s(\tau) = T - 2 \times |\{i \mid s_i \neq s_{i+\tau}\}|$$

Ainsi,  $C_s(\tau) = T$  si et seulement si  $\tau$  est un multiple de la période.

# Les critères de Golomb (1982)

Une suite binaire  $(s_n)_{n \in \mathbb{N}}$  périodique de période  $T$  pseudoaléatoire doit vérifier les critères suivants :

- **Équirépartition des runs** : Soit  $S$  l'ensemble des runs de  $(s)$ , et soit  $k$  tel que  $2^k \leq |S| < 2^{k+1}$ . Alors  $(s)$  doit avoir  $|S|/2^\ell$  runs de taille  $\ell$  pour  $1 \leq \ell \leq k$ , et le nombre de runs de 0 doit différer du nombre de runs de 1 d'au plus 1.
- **Suite équilibrée** (Conséquence) : Sur une période, le nombre de 0 et de 1 diffère au plus de 1 :

$$\left| \sum_{i=0}^{T-1} (-1)^{s_i} \right| \leq 1$$

- **Faible autocorrélation** :  $C_s$  est constante sur  $\mathbb{Z} \setminus T\mathbb{Z}$ ,  
i.e.  $(s)$  est indépendante de tous ses translatés.

# Quelques références



Solomon W. Golomb. - *Shift Register Sequences* - Aegean Park Press, 1982.



Tor Helleseth - *Maximal-Length Sequences* - In *Encyclopedia of Cryptography and Security*, 2nd Ed., 2011.

# Un exemple

## Exercice

Vérifiez que la suite de période 31 :

1,0,0,0,0,1,0,1,0,1,1,1,0,1,1,0,0,0,1,1,1,1,1,0,0,1,1,0,1,0,0

vérifie les critères de Golomb.

## Remarques :

- C'est un exemple de suite produite par un LFSR (*Linear Feedback Shift Register* ou Registre à décalage à rétroaction linéaire) de longueur 5, et dont le polynôme de rétroaction est  $X^5 + X^2 + 1$ .
- On verra qu'une telle suite n'est **pas imprédictible**, mais est à la base de nombreux stream ciphers.

# Générateurs Pseudoaléatoires à la Sécurité Prouvée ?

## Ouverture : Cryptographie Avancée - **Sécurité réductionniste.**

- Pour le moment : Sécurité assurée par la cryptanalyse *ad-hoc*.
- Il existe des PRG dont la sécurité repose sur des problèmes bien connus :
  - Le PRG de Blum, Blum et Shub (1986) est **impredictible** si le problème de *résidu quadratique* est difficile (*a priori* plus simple que la factorisation).
  - Le PRG de Blum et Micali (1984) est **imprédictible** si le problème du logarithme discret dans  $(\mathbb{Z}/p\mathbb{Z})^\times$  est difficile. En particulier,  $p$  doit être grand !

Ces deux PRG sont très lents.

# Linear Feedback Shift Registers

# Linear Feedback Shift Registers (LFSR)

## Définition

Un LFSR de longueur  $\ell$  est une machine à états finis qui produit une suite  $(s_t)_{t \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$  satisfaisant une relation de récurrence linéaire d'ordre  $\ell$  sur  $\mathbb{F}_q$  :

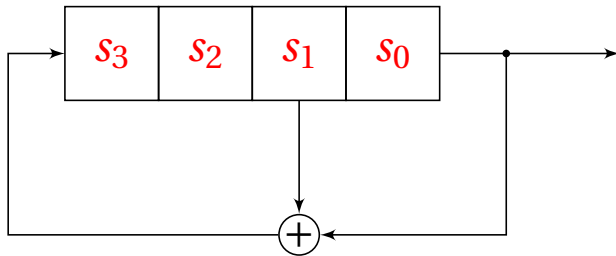
$$s_{t+\ell} = c_1 s_{t+\ell-1} + \cdots + c_\ell s_t, \quad \forall t \geq 0.$$

où les coefficients  $c_1, \dots, c_\ell \in \mathbb{F}_q$  sont fixés.

## Proposition

Toute suite récurrente linéaire d'ordre  $\ell$  est produite par un LFSR de longueur  $\ell$ .

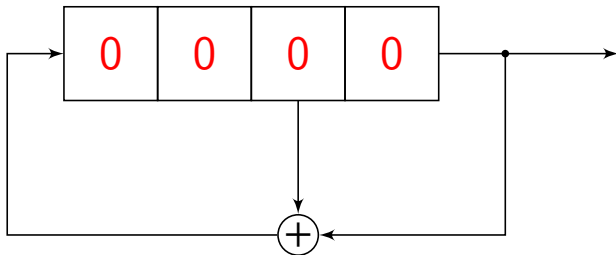
## Exemple : LFSR binaire



LFSR binaire avec coefficients  $(c_1, c_2, c_3, c_4) = (0, 0, 1, 1)$

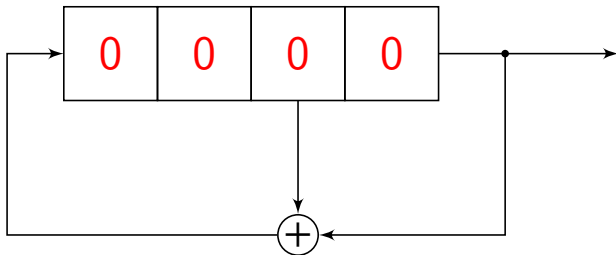
Implémente la récurrence  $s_{t+4} = s_{t+1} + s_t$

## Exemple : LFSR binaire



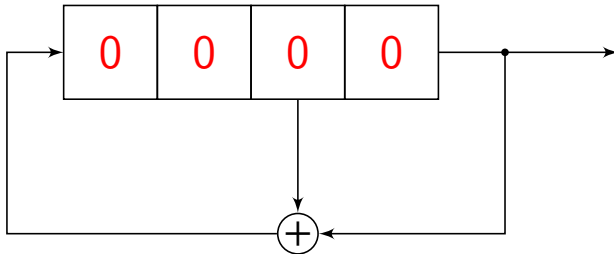
0

## Exemple : LFSR binaire



00

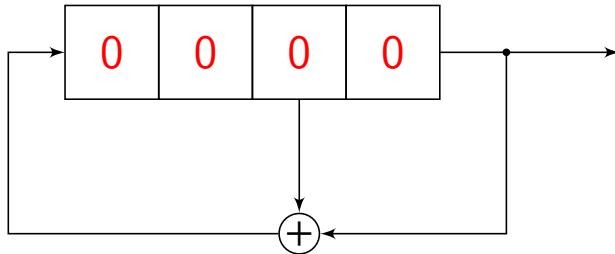
## Exemple : LFSR binaire



000

Il faut éliminer l'état 0.

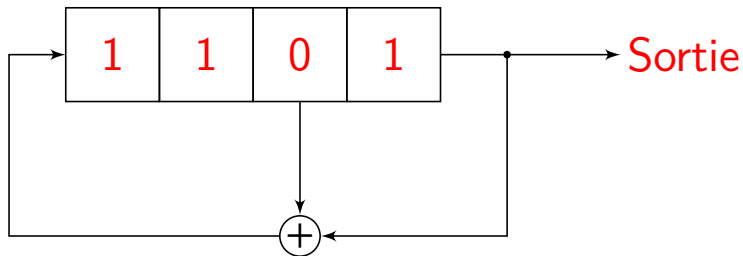
## Exemple : LFSR binaire



0000...

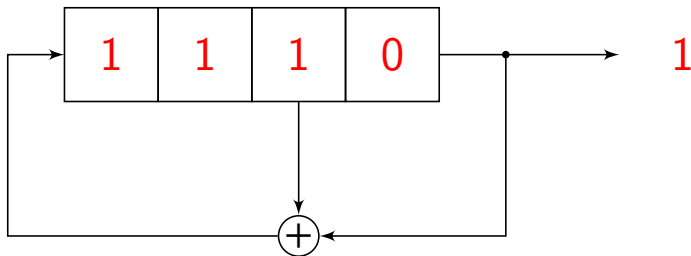
Il faut éliminer l'état 0.

## Exemple : LFSR binaire non trivial



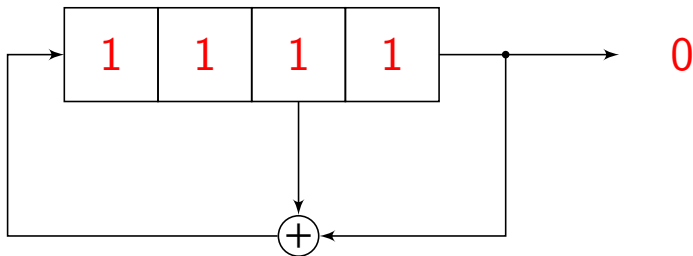
État Initial.

## Exemple : LFSR binaire non trivial



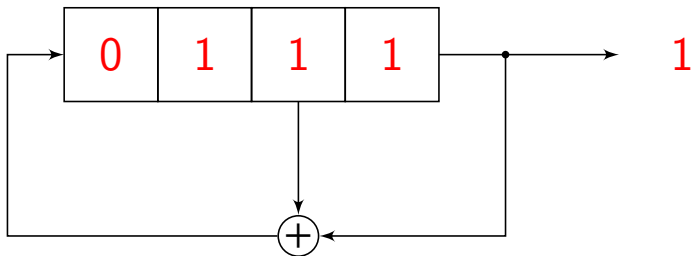
1

## Exemple : LFSR binaire non trivial



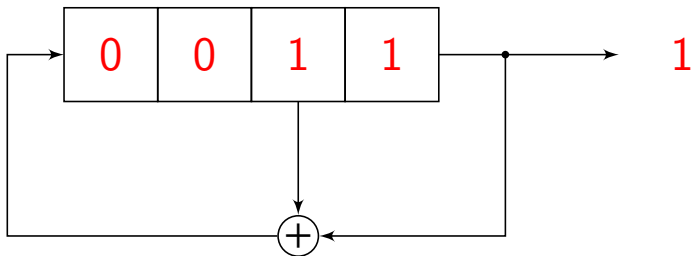
10

## Exemple : LFSR binaire non trivial



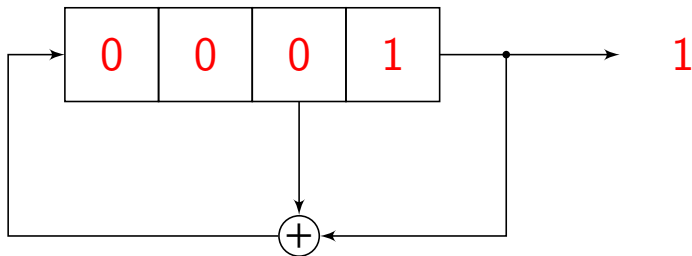
101

## Exemple : LFSR binaire non trivial



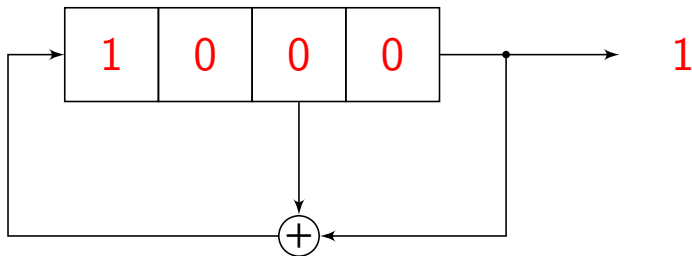
1011

## Exemple : LFSR binaire non trivial



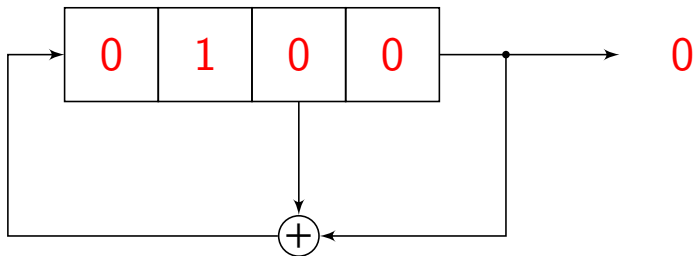
10111

## Exemple : LFSR binaire non trivial



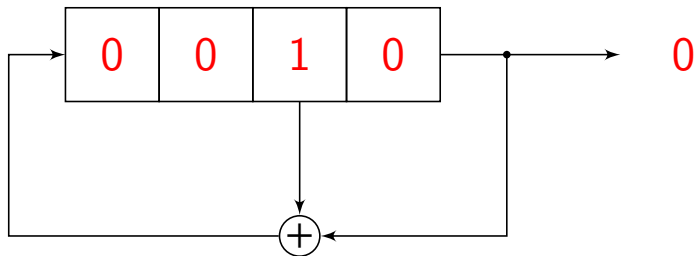
101111

## Exemple : LFSR binaire non trivial



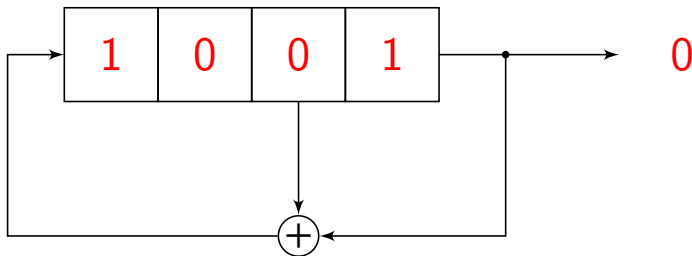
1011110

## Exemple : LFSR binaire non trivial



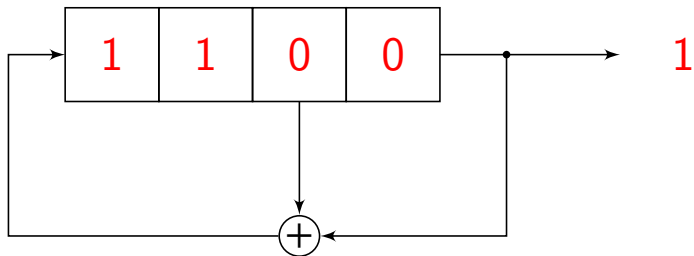
10111100

## Exemple : LFSR binaire non trivial



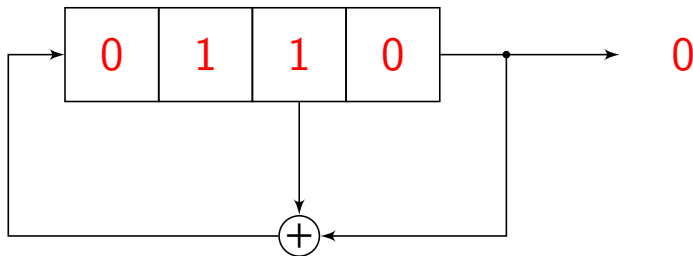
101111000

## Exemple : LFSR binaire non trivial



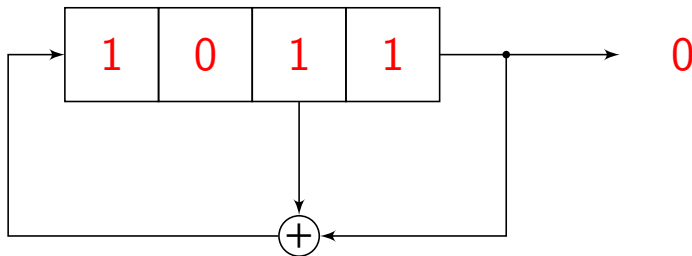
1011110001

## Exemple : LFSR binaire non trivial



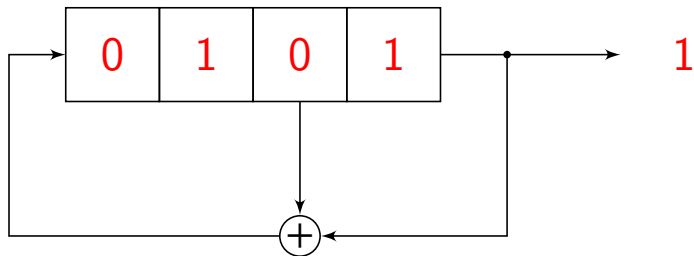
10111100010

## Exemple : LFSR binaire non trivial



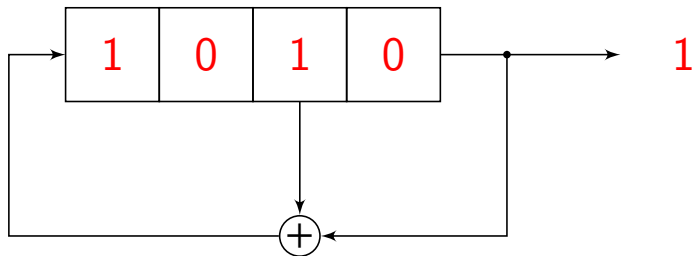
101111000100

## Exemple : LFSR binaire non trivial



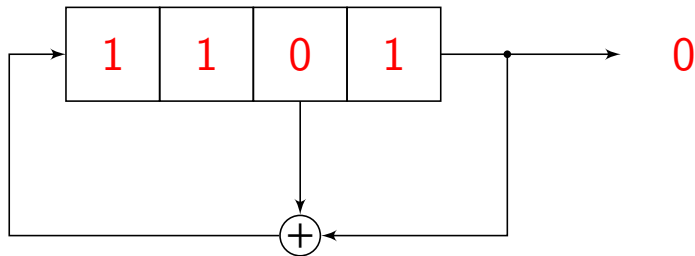
1011110001001

## Exemple : LFSR binaire non trivial



10111100010011

## Exemple : LFSR binaire non trivial



101111000100110...

La suite produite est périodique, de période  $15 = 2^4 - 1$ .

# Polynôme de Rétroaction

On représente souvent les coefficients d'un LFSR sous la forme d'un polynôme appelé **polynôme de rétroaction** du LFSR, défini par

$$P(X) \stackrel{\text{def}}{=} 1 - \sum_{i=1}^{\ell} c_i X^i = - \sum_{i=0}^{\ell} c_i X^i \in \mathbb{F}_q[X] \quad \text{avec la convention } c_0 = -1.$$

Un LFSR de longueur  $\ell$  est dit non-singulier si  $c_\ell \neq 0$ , *i.e.* si son polynôme de rétroaction est de degré  $\ell$ .

## Exemple

Le LFSR précédent a pour polynôme de rétroaction

$$P(X) = 1 + X^3 + X^4 \in \mathbb{F}_2[X]$$

# Une remarque importante

Soit  $\ell$  la taille du registre. L'état interne peut alors prendre **au plus**  $2^\ell$  valeurs différentes, et la suite  $(s_t)$  est donc **ultimement périodique** de période au plus  $2^\ell$ .

On va chercher à produire des suites chiffrantes de **période maximale**.

Propriétés d'une suite produite par un LFSR

# Séries Formelles

L'espace des suites  $\mathbb{F}_q^{\mathbb{N}}$  est muni d'une structure de  $\mathbb{F}_q$ -algèbre (de dimension infinie) :

$$(a_t) \odot (b_t) \stackrel{\text{def}}{=} (c_t)$$

où

$$c_t \stackrel{\text{def}}{=} \sum_{k=0}^t a_k b_{t-k}.$$

Cette structure est appelée algèbre des **séries formelles** à une indéterminée sur  $\mathbb{F}_q$  et est dénotée  $\mathbb{F}_q[[X]]$ . Une suite  $(a_t)$  vue comme élément de  $\mathbb{F}_q[[X]]$  est notée

$$\sum_{t \geq 0} a_t X^t$$

aussi appelée **série génératrice** de  $(a_t)$ .

# Quelques Propriétés et Exemples

- $\mathbb{F}_q[[X]]$  est une algèbre commutative, intègre, de dimension infinie sur  $\mathbb{F}_q$ .
- $\mathbb{F}_q[[X]]$  contient l'algèbre des polynômes  $\mathbb{F}_q[X]$ .
- $A = \sum_{t \geq 0} a_t X^t \in \mathbb{F}_q[[X]]$  est inversible si et seulement si  $a_0 \neq 0$ .
- $\mathbb{F}_q[[X]]$  n'est pas un corps.

- $1 + X \in \mathbb{F}_q[X] \subset \mathbb{F}_q[[X]]$  est inversible, d'inverse

$$\frac{1}{1+X} \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} (-1)^t X^t$$

- $X$  n'est pas inversible dans  $\mathbb{F}_q[[X]]$ .

# Longueur d'une suite LFSR

Pour  $(s_t) \in \mathbb{F}_q^{\mathbb{N}}$ , on note  $S(X) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} s_t X^t$  sa série formelle associée.

## Théorème

Une suite  $(s_t)$  est produite par un LFSR de longueur  $\ell$  et de polynôme de rétroaction  $P$  si et seulement si il existe un polynôme  $Q(X)$  avec  $\deg Q < \ell$  et tel que

$$S(X) = \frac{Q(X)}{P(X)}.$$

De plus,  $Q(X)$  est entièrement déterminé par les coefficients de  $P$  et par l'état initial :

$$Q(X) = - \sum_{k=0}^{\ell-1} X^k \sum_{i=0}^k c_i s_{k-i}$$

# Preuve

On rappelle la définition de  $P$

$$P(X) = 1 - \sum_{i=1}^{\ell} c_i X^i = - \sum_{i=0}^{\ell} c_i X^i = \sum_{i=0}^{\infty} (-c_i \mathbb{1}_{i \leq \ell}) X^i$$

Alors,

$$P(X) \cdot S(X) = \sum_{k=0}^{\infty} X^k \sum_{i=0}^k (-c_i \mathbb{1}_{i \leq \ell}) s_{k-i}$$

On découpe la somme selon que  $k < \ell$  ou  $k \geq \ell$  :

$$P(X) \cdot S(X) = - \underbrace{\sum_{k=0}^{\ell-1} X^k \sum_{i=0}^k c_i s_{k-i}}_{\stackrel{\text{def}}{=} Q(X) \in \mathbb{F}_q[X]} + \sum_{k \geq \ell} X^k \sum_{i=0}^{\ell} -c_i s_{k-i}.$$

## Preuve (Cont'd)

Il suffit de prouver que les coefficients des termes d'ordre supérieurs à  $\ell$  sont tous nuls.  
Pour cela, on écrit  $k = t + \ell$  avec  $t \geq 0$  :

$$\sum_{k \geq \ell} X^k \sum_{i=0}^{\ell} -c_i s_{k-i} = \sum_{t=0}^{\infty} X^{t+\ell} \sum_{i=0}^{\ell} -c_i s_{t+\ell-i}$$

*i.e.*

$$\sum_{k \geq \ell} X^k \sum_{i=0}^{\ell} -c_i s_{k-i} = \sum_{t=0}^{\infty} X^{t+\ell} \left( \underbrace{s_{t+\ell} - (c_1 s_{t+\ell-1} + \cdots + c_{\ell} s_t)}_{=0 \text{ (relation de récurrence)}} \right)$$

# Conséquence 1 : Sparsification

## Proposition

Toute suite produite par un LFSR de rétroaction  $P$  peut aussi être produite par tout LFSR de rétroaction un multiple non nul de  $P$ .

**Exemple :** Soit  $(s_t)$  une suite binaire vérifiant

$$s_{t+6} = s_{t+4} + s_{t+3} + s_{t+1} + s_t, \quad \forall t \geq 6.$$

Son polynôme de rétroaction est donc  $P(X) = 1 + X^2 + X^3 + X^5 + X^6$ . On vérifie que cette suite vérifie aussi  $s_{t+8} = s_{t+7} + s_t$ , puisque

$$1 + X + X^8 = (1 + X + X^2)P(X).$$

Si un multiple de  $P$  est moins dense, la récurrence est moins coûteuse à implémenter !

## Conséquence 2 : Minimisation

Soit  $s \stackrel{\text{def}}{=} (s_t)_{t \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$  une suite récurrente linéaire non nulle.

Il existe un unique polynôme  $P_0$  de terme constant 1 tel que la série génératrice  $S(X)$  soit de la forme

$$S(X) = \sum_{t \geq 0} s_t X^t = \frac{Q_0(X)}{P_0(X)}$$

où  $\gcd(P_0, Q_0) = 1$ . Le degré de  $P_0$  est alors **minimal** parmi celui de tous les polynômes de rétroaction générant la même suite  $(s_t)$ .

### Exercice

Soit  $s$  la suite définie par l'état initial  $(s_0, \dots, s_9) = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1)$  et de polynôme de rétroaction  $P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$ .

Vérifier que  $s$  est en réalité générée par un LFSR de longueur 3.

# Complexité Linéaire d'une Suite

## Définition

La **complexité linéaire**  $\Lambda(s)$  d'une suite  $(s_t)_{t \in \mathbb{N}}$  est :

- 0 si  $s$  est la suite nulle.
- le degré de son polynôme de rétroaction minimal s'il existe.
- $\infty$  sinon.

Lorsque le polynôme de rétroaction  $P$  est irréductible, la complexité linéaire est **exactement** le degré de  $P$ . Pour les applications cryptographiques, on choisit donc toujours  $P$  irréductible.

Soit  $(s_t)$  une suite récurrente linéaire de complexité linéaire  $\Lambda$ . Il suffit de  $2\Lambda$  termes consécutifs pour déterminer entièrement la suite.

# Période d'un LFSR

Soit  $s \stackrel{\text{def}}{=} (s_t)_{t \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$  une suite récurrente linéaire, et soit  $P_0$  son polynôme de rétroaction **minimal**.

- $(s_t)$  est périodique, de période  $T \leq 2^{\deg P_0} - 1$ .
- La plus petite période est égale au plus petit entier  $e$  tel que  $P(X)$  divise  $X^e + 1$ .
- En particulier, la période est **maximale**, égale à  $2^{\deg P_0} - 1$  si et seulement si  $P_0$  est un polynôme **primitif**.

## Exemple

Le polynôme de rétroaction du LFSR du début était  $P(X) = 1 + X^3 + X^4$  qui est bien primitif. On retrouve que sa période est bien de  $2^4 - 1 = 15$ .

# LFSR et Statistiques

Une suite produite par un LFSR d'état initial non nul et de polynôme de rétroaction primitif satisfie les critères statistiques de Golomb.

## Exemple

Dans l'exemple du début, le registre interne prend toutes les valeurs de  $\mathbb{F}_2^4 \setminus \{0\}$ . En particulier, le premier élément de chaque état prend  $2^3 = 8$  fois la valeur 1 et  $2^3 - 1 = 7$  fois la valeur 0. La suite chiffrente est exactement la suite de ces premiers éléments.

# Algorithme de Berlekamp-Massey

Soit  $(s_t)$  une suite récurrente linéaire de complexité linéaire  $\Lambda(s)$  inconnue.

Il existe un algorithme très efficace qui détermine le polynôme de rétroaction minimal à partir de n'importe quelle sous-séquence de  $2\Lambda(s)$  termes consécutifs.

- Pour avoir des bonnes garanties de sécurité, il faudrait que la complexité linéaire soit extrêmement grande afin de résister à l'attaque de Berlekamp-Massey. Cependant, ceci implique d'avoir des LFSR de longueur trop grosse pour que ça soit praticable.
- Un LFSR seul n'est en général **pas suffisant** pour apporter de la sécurité. On verra au cours 3 comment **augmenter** la complexité linéaire afin de résister à l'attaque de Berlekamp-Massey.