

# Jeu de Marienbad, codage, cryptographie et stéganographie élémentaires

Daniel Augot

INRIA Saclay-Île de France et École polytechnique

Séminaire CINÉ  
École polytechnique, novembre 2010

# Plan

Jeu de Marienbad

Addition modulo 2

Applications : codes binaires

Applications : stéganographie

Application : chiffrement à flot

*coder, décoder, chiffrer, déchiffrer, décrypter* mais pas crypter !

# Le jeu de Marienbad



L'année dernière à Marienbad (1961). Réalisateur Alain Resnais.  
Scénario Alain Robbe-Grillet. Avec Delphine Seyrig.

# Le jeu de Marienbad



## Règle

- ▶ À tour de rôle, chacun retire des allumettes dans un même tas.
- ▶ Version misère : celui qui retire la dernière allumette a perdu.
- ▶ Version normale : celui qui retire la dernière allumette a gagné.

# Codage de l'état stratégique d'un jeu

1. On écrit les tailles des tas en base 2 ;
2. On fait la somme, sans tenir compte des retenues. C'est la *somme-nim*. Ou XOR.

## Exemple



Tas	Taille	$2^2$	$2^1$	$2^0$	
1	1	0	0	1	
2	3	0	1	1	
3	5	1	0	1	
4	7	1	1	1	
		0	0	0	<i>somme-nim</i>

**Lemme.** Si la somme-nim est nulle, quelque soit le mouvement suivant, la somme-nim devient non nulle.

## Théorème (Version normale)

*Si la somme-nim initiale est non nulle, le premier joueur a une stratégie gagnante.*

## Exemple

- ▶ État



$$S = 3 \oplus 4 \oplus 5 = 2 \neq 0.$$

- ▶ On calcule

$$3 \oplus S = 1$$

$$4 \oplus S = 6$$

$$5 \oplus S = 7$$

Le seul tas dont la somme a diminué est le tas 1.

- ▶ On enlève deux allumettes dans le tas 1.

$$\begin{aligned} S &= T_1 \oplus T_2 \oplus T_3 \\ 0 = S \oplus S &= S \oplus (T_1 \oplus T_2 \oplus T_3) \\ 0 &= (T_1 \oplus S) \oplus T_2 \oplus T_3 \end{aligned}$$

**Lemme.** Il existe forcément un tas dont la taille diminue.

## Pseudo-code et invariant de boucle

tant que Il reste une allumette faire

- *Au tour de la machine. La somme de nim est non nulle*

Calculer la somme de nim  $S$  du jeu

pour chaque tas  $i$  faire

$T_i \leftarrow$  la taille du tas  $i$

si  $T_i \oplus S < T_i$  alors

$T_i \leftarrow T_i \oplus S$

Interrompre la boucle

fin si

fin pour

- *Au tour de l'adversaire. La somme de nim est nulle*

// Quelque soit son mouvement,

// la somme de Nim deviendra non nulle

fin tant que

# Construction par induction d'un corps sur $\mathbb{N}$

## Définition (Mex)

Pour  $I \subset \mathbb{N}$ ,  $I \neq \mathbb{N}$  :  $\text{mex}(I) = \min(\mathbb{N} \setminus I)$ .

## Définition (Somme de Nim, $\oplus$ )

$$i \oplus j = \text{mex} \left( \{i' \oplus j; i' < i\} \cup \{i \oplus j'; j' < j\} \right)$$

## Définition (Produit de Nim, $\otimes$ )

$$i \otimes j = \text{mex} \{ (i' \otimes j) \oplus (i \otimes j') \oplus (i' \otimes j'); i' < i, j' < j \}$$

Justification :

$$(i - i') \otimes (j - j') > 0 \implies i \otimes j > (i' \otimes j) \oplus (i \otimes j') \oplus (i' \otimes j')$$



# Tables

$\oplus$  et  $\otimes$

$$i \oplus j = \text{mex} (\{i' \oplus j; i' < i\} \cup \{i \oplus j'; j' < j\})$$

$$i \otimes j = \text{mex} \{(i' \otimes j) \oplus (i \otimes j') \oplus (i' \otimes j'); i' < i, j' < j\}$$

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$\otimes$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

## Résultat : les « nimbers »

### Théorème

*On obtient tous les corps finis de taille  $\mathbb{F}_{2^{2^n}}$ .*

Définition « canonique » de  $\mathbb{F}_{2^{2^n}}$ . Par exemple  $\mathbb{F}_{256}$  (les octets).

### Théorème

*On obtient un corps infini de caractéristique 2.*

Ce corps est la « clôture quadratique de  $\mathbb{F}_2$  ».

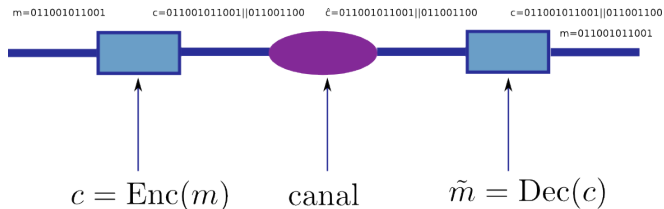
### Théorème

*Tout polynôme de degré  $2^{2^n}$  à coefficients dans ce corps y est scindé.*

John H. Conway. *On Numbers and Games*. 2nd edition, 2000

H.W. Lenstra Jr. Nim multiplication. Technical report, Institut des Hautes Études Scientifiques, 1978

# Codage dans un canal de transmission



## Définition

Le **code** correspondant à la fonction d'encodage **Enc** est l'ensemble des mots codés, i.e.

$$C = \{c = \text{Enc}(m); \quad m \in \mathbb{F}_2^k\}$$

## Deux extrêmes triviaux, code de parité, code de répétition

### Code à parité

Encodage Enc

$$(x_1, \dots, x_n) \mapsto (x_1, \dots, x_n, x_{n+1})$$

avec  $x_{n+1} = \bigoplus_{i=1}^n x_i$ .

- ▶ redondance minimale, i.e. taux de transmission maximal ;
- ▶ détecte une erreur, pas de correction.

### Code de répétition

Encodage Enc

$$(x_1) \mapsto \underbrace{(x_1, \dots, x_1)}_{n \text{ fois}}$$

- ▶ redondance maximale, i.e. taux de transmission minimal.
- ▶ taux de correction maximal (vote majoritaire).

# Distance de Hamming

## Définition (Distance de Hamming)

La distance de Hamming entre  $x$  et  $y$  est

$$d(x, y) = |\{i; x_i \neq y_i\}|.$$

Le poids de Hamming de  $x$  est  $d(x, 0)$ .

## Définition

Soit  $C \subset \mathbb{F}_2^n$ , la distance minimale de  $C$  est

$$d(C) = \min_{x, y \in C, x \neq y} d(x, y).$$

Si le code est  $\mathbb{F}_2$ -linéaire, alors

$$d(C) = \min_{x \neq 0 \in C} w(x).$$

# Hamming 1950

- ▶ 4 bits d'information  $x_1, x_2, x_3, x_4$ , trois bits de redondance  $x_5, x_6, x_7$  ;
- ▶ trois *équations de « parité »* :

$$x_5 = x_1 \oplus x_2 \oplus x_4$$

$$x_6 = x_1 \oplus x_3 \oplus x_4$$

$$x_7 = x_2 \oplus x_3 \oplus x_4$$

- ▶ *matrice de parité* :

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

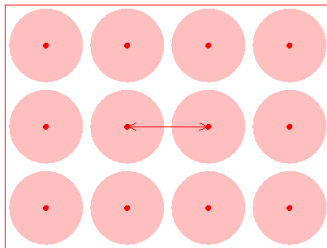
telle que

$$c \in C \iff H \cdot c^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} ;$$

- ▶ les colonnes sont deux à deux distantes  $\implies$  pas de mots de poids 2 dans le code ; le poids minimal est trois.

## Distance minimale et capacité de correction

- ▶ Si un code a pour distance minimale  $d$ , alors les boules de rayon  $t = \lfloor \frac{d-1}{2} \rfloor$  sont disjointes.



- ▶ On note alors  $t = \lfloor \frac{d-1}{2} \rfloor$ , c'est la *capacité de correction du code*.

## Décodage

- ▶ Le code de Hamming est de distance minimale 3 ;
- ▶ il est donc 1-correcteur ;

### Décodage :

- ▶ Soit  $y = c \oplus e$ . Pour une erreur  $e$  de poids 1, par exemple

$$e = (0, 0, 0, 1, 0, 0, 0)$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} e^T = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

### Algorithme :

Calculer  $S = Hy^T = H(c \oplus e)^T = He^T$  ;

Identifier  $S$  comme une colonne de la matrice de parité ;

L'indice de cette colonne donne la position du bit à inverser.

// *Algorithme propre au code de Hamming.*



# Généralisation

Une infinité de codes de Hamming :

- ▶ Pour  $m$  donné, on liste tous les entiers non nuls entre 1 et  $2^m - 1$  ;
- ▶ On met en colonne les représentations binaires de ces entiers. Matrice à  $m$  lignes.
- ▶ C'est la matrice de parité du code de Hamming de longueur  $2^m - 1$  ;
- ▶ On obtient un code de distance minimale trois.

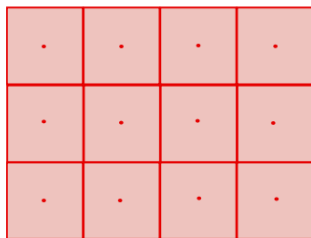
## Note

*Pour  $d$ ,  $k$ , et  $n$  généraux, on ne connaît pas le rapport entre la dimension  $k$  et la distance minimale  $d$ .*

## Le code de Hamming est « parfait »

{Les boules de rayon 1}  $\times$  {les mots de code} = tout l'espace

$$(1 + (2^m - 1)) 2^{2^m - 1 - m} = 2^{2^m - 1}$$



La distance de Hamming est non intuitive !

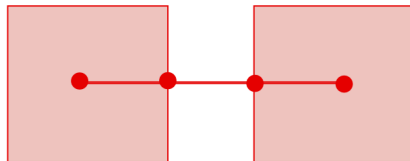
## NVidia Fermi (2009)

- ▶ La carte Fermi implémente des codes *SECDED* « Single-Error Correct Double-Error Detect ».
- ▶ Construction à base des codes de Hamming :
  1. on considère un code de Hamming de longueur  $127 = 2^7 - 1$ , de dimension  $127 - 7$ , distance minimale 3 ;
  2. on l'étend en ajoutant un « bit de parité » : code de longueur 128, distance minimale 4 ; redondance 8 bits.
  3. on le raccourcit pour obtenir un code de longueur 72, de dimension  $72 - 8$ , de distance minimale 4.
- ▶ Sont protégés :
  1. les registres ;
  2. la mémoire partagée ;
  3. les caches L1 et L2.

Même redondance que 8 codes de parité sur  $8 \times 7$  bits (8 bits), mais correction d'erreur.

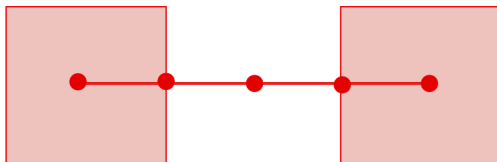
## Pourquoi distance 4 ?

- ▶ Hamming standard ( $d = 3$ ) :



Une erreur de poids 2 est mal corrigée.

- ▶ Hamming étendu ( $d = 4$ ) :



Une erreur de poids 2 est détectée.

Seulement en ajoutant un bit de parité !

# Stéganographie (F5)

- ▶ But : cacher un secret  $s$  dans une « couverture »  $x$  (image, vidéo, son).
- ▶ La plupart des médias perceptifs tolèrent des modifications mineures dans leur représentation binaire, sans que ce soit perceptible.
- ▶ Exemple naïf : les bits les moins significatifs d'une représentation binaire d'une image, par octets par exemple (format ppm, en RGB).
- ▶ Exemple plus sophistiqué : les bits de poids faible de la discrétisation de la transformée en cosinus de l'image (JPEG).
- ▶ Algorithme célèbre : F5.

## Modélisation discrète

- ▶ passage d'un modèle « signal », ou « perceptif », à un modèle « discret » ;
- ▶ la *couverture* est  $(x_1, \dots, x_n) \in \mathbb{F}_2^n$  ;
- ▶ le *secret* est  $(s_1, \dots, s_m) \in \mathbb{F}_2^m$  ; et  $m \ll n$  ;
- ▶ On cherche un schéma *plongement-extraction*  $(E, R)$  :

$$E : \mathbb{F}_2^m \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \quad \text{et} \quad R : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$$

tel que

$$\forall x, \quad R(E(s, x)) = s \quad (\text{correction de la méthode})$$

et

$$\rho = \max_x d(x, E(s, x)) \quad \text{est petit (furtivité)}$$

## Stéganographie (F5 et code de Hamming)

- ▶ Le code de Hamming  $[n = 2^m - 1, 2^m - 1 - m, 3]_2$  est *parfait* :
- ▶ Pour tout secret  $s \in \mathbb{F}_2^m$ , il existe un mot de poids 1 tel que

$$H_m e^T = s.$$

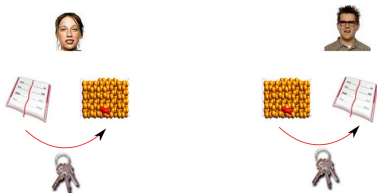
- ▶ Donc si la couverture est de longueur  $n = 2^m - 1$ , on peut y plonger  $m$  bits, en modifiant 1 bit de la couverture.
- ▶ Logique : un mot de poids 1 de longueur  $2^m - 1$  peut être vu comme un entier  $i \in [0, 2^m - 1]$ , donc comme  $m$  bits.

### Exemple

On peut cacher 10 bits dans  $2^{10} = 1024$  bits en ne modifiant qu'un bit de ceux-ci.

Problème : on ne passe si facilement à plus de bits d'information.

# Chiffrement symétrique



On veut donc un algorithme de chiffrement :

$$f_C : \mathbb{F}_2^k \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \\ (K, M) \mapsto C = f_C(K, M)$$

et un algorithme de déchiffrement

$$f_D : \mathbb{F}_2^k \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n \\ (K, C) \mapsto M = f_D(K, C)$$



# Chiffrement de César, Masque jetable, chiffre de Vernam, etc

- ▶ La clé est  $K = (K_1, \dots, K_n) \in \mathbb{F}_2^n$ ;
- ▶ Le message est  $M = (M_1, \dots, M_n) \in \mathbb{F}_2^n$ ;
- ▶ Le chiffré est  $C = (C_1, \dots, C_n) \in \mathbb{F}_2^n$ , avec

$$C_i = M_i \oplus K_i, \quad i \in \{1, \dots, n\}$$

- ▶ Pour déchiffrer, on fait

$$C_i \oplus K_i = M_i, \quad i \in \{1, \dots, n\}$$

## Théorème

*Si la clé est aléatoire, c'est-à-dire si :*

$$p(K_i = 1) = \frac{1}{2},$$

*alors la connaissance de C ne donne aucune information sur M.*

# Chiffrement de César, Masque jetable, chiffre de Vernam

## Théorème

*Si la clé est aléatoire, c'est-à-dire si :*

$$p(K_i = 1) = \frac{1}{2},$$

*alors la connaissance de  $C$  ne donne aucune information sur  $M$ .*

## Démonstration.

Supposons  $C_i = M_i \oplus K_i = 0$ , alors

$$p(M_i = 0) = p(K_i = 0) = \frac{1}{2}. \quad \square$$

*À condition que l'alea soit parfait.*

*À condition que la clé soit aussi longue que le message.*

*À condition que la clé ne soit utilisée qu'une fois.*

# Générateur pseudo-aléatoire

Il est donné par :

- ▶ une « graine » courte,  $s_0 \in \mathbb{F}_2^m$  ;
- ▶ une fonction de mise à jour :  $U : \mathbb{F}_{2^m} \rightarrow \mathbb{F}_{2^m}$  ;
- ▶ une fonction d'extraction :  $E : \mathbb{F}_{2^m} \rightarrow \mathbb{F}_2$ .

Pour chaque bit  $M_i$  à chiffrer,  $s_i$  étant l'état à l'étape  $i$  :

1. extraire un bit  $K_i = E(s_i)$
2. chiffrer :  $C_i = M_i \oplus K_i$  ;
3. mettre à jour l'état :  $s_{i+1} = U(s_i)$ .

Le générateur pseudo-aléatoire doit

1. passer un certains nombres de tests statistiques (NIST) ;
2. être difficile à inverser (impossible de retrouver  $s_0$ ) ;
3. impossible à prédire, même après observation.

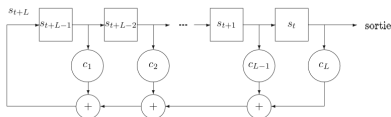
Est-ce qu'il en existe vraiment ?

# Registre à décalage à rétroaction linéaire

- ▶ l'état initial est  $S_1, \dots, S_m$  ;
- ▶ on fixe des constantes  $(c_1, \dots, c_m) \in \mathbb{F}_2^m$

$$S_{t+m} = c_1 S_{t+m-1} \oplus \dots \oplus c_m S_t; \quad t \geq m + 1$$

- ▶ extraction :  $E(S_{t+m-1}, \dots, S_t) = S_t$  ;
- ▶ implantation matérielle très simple :



- ▶ très bonnes propriétés statistiques ;
- ▶ Mais facile à prédire :

## Proposition

*L'observation de  $2m$  bits de la séquence produite permet de retrouver la rétroaction et l'initialisation. Complexité  $O(m^2)$ .*

# Conclusion

- ▶ Une définition naturelle pour la somme de Nim des entiers ;
- ▶  $1 \oplus 1 = 0$  est fondamental en informatique, parce que c'est simple !
- ▶ force de la simplicité du code de Hamming et son décodage ;
- ▶ application en stéganographie ;
- ▶ aléa déterministe cryptographique : niveau recherche.

