# Codes correcteurs

Clément PERNET
clement.pernet@univ-grenoble-alpes.fr

M1 AM Cryptology Complement

# Sommaire

# Sommaire

# Principle of the evaluation-interpolation codes

### Theorem (Interpolation)

*For all distinct* $x_1, \ldots, x_k$ *and all* $y_1, \ldots, y_k$*, there is a unique polynomial* $f = f_0 + f_1 x + \ldots f_{k-1} x^{k-1}$ *with degree* $< k$ *such that :*

$$f(x_j) = y_j, \text{ for all } 1 \leq j \leq k.$$

# Principle of the evaluation-interpolation codes

## Theorem (Interpolation)

*For all distinct* $x_1, \ldots, x_k$ *and all* $y_1, \ldots, y_k$, *there is a unique polynomial* $f = f_0 + f_1 x + \ldots f_{k-1} x^{k-1}$ *with degree* $< k$ *such that :*

$$f(x_j) = y_j, \text{ for all } 1 \leq j \leq k.$$

## Corollary

*For fixed* $x_i$*'s*

- *equivalent representation :* $(y_1, \ldots, y_k) \Leftrightarrow (f_0, \ldots, f_{k-1})$.
- *over-sampling :* $(y_1, \ldots, y_k, y_{k+1}, \ldots, y_n) \Leftarrow (f_0, \ldots, f_{k-1})$.
  $\Rightarrow$ *redundancy added*

# Reed-Solomon codes

## Definition (Reed-Solomon codes)

Over a finite field $K$, let $x_1, \dots, x_n \in K$ distinct elements. The Reed-Solomon code of length $n$ and dimension $k$ is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \dots, f(x_n)), f \in K[X]; \deg f < k\}$$

# Reed-Solomon codes

## Definition (Reed-Solomon codes)

Over a finite field $K$, let $x_1, \ldots, x_n \in K$ distinct elements. The Reed-Solomon code of length $n$ and dimension $k$ is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \ldots, f(x_n)), f \in K[X]; \deg f < k\}$$

## Example

$(n,k) = (5,3)$ over $\mathbb{Z}/19\mathbb{Z}$. $(x_1, x_2, x_3, x_4, x_5) = (1, 5, 8, 10, 12)$
Message : $(1,2,1) \in (\mathbb{Z}/19\mathbb{Z})^3 \to f(X) = 1 + 2x + x^2$
$(1,2,1) \xrightarrow{Eval} (f(1), f(5), f(8), f(10), f(12)) = (4, 17, 5, 7, 17)$

# Reed-Solomon codes

## Definition (Reed-Solomon codes)

Over a finite field $K$, let $x_1, \ldots, x_n \in K$ distinct elements. The Reed-Solomon code of length $n$ and dimension $k$ is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \ldots, f(x_n)), f \in K[X]; \deg f < k\}$$

## Example

$(n,k) = (5,3)$ over $\mathbb{Z}/19\mathbb{Z}$. $(x_1, x_2, x_3, x_4, x_5) = (1, 5, 8, 10, 12)$
Message : $(1,2,1) \in (\mathbb{Z}/19\mathbb{Z})^3 \to f(X) = 1 + 2x + x^2$
$(1,2,1) \xrightarrow{Eval} (f(1), f(5), f(8), f(10), f(12)) = (4, 17, 5, 7, 17)$
$(4, 17, 5, 7, 17) \xrightarrow{Interp.} (1, 2, 1, 0, 0)$ $\qquad x^2 + 2x + 1$

# Reed-Solomon codes

## Definition (Reed-Solomon codes)

Over a finite field $K$, let $x_1, \ldots, x_n \in K$ distinct elements. The Reed-Solomon code of length $n$ and dimension $k$ is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \ldots, f(x_n)), f \in K[X]; \deg f < k\}$$

## Example

$(n,k) = (5,3)$ over $\mathbb{Z}/19\mathbb{Z}$. $(x_1, x_2, x_3, x_4, x_5) = (1, 5, 8, 10, 12)$
Message : $(1,2,1) \in (\mathbb{Z}/19\mathbb{Z})^3 \to f(X) = 1 + 2x + x^2$
$(1,2,1) \xrightarrow{Eval} (f(1), f(5), f(8), f(10), f(12)) = (4, 17, 5, 7, 17)$
$(4, 17, 5, 7, 17) \xrightarrow{Interp.} (1, 2, 1, 0, 0)$ $\qquad x^2 + 2x + 1$
$(4, 17, 13, 7, 17) \xrightarrow{Interp.} (12, 8, 11, 10, 1)$ $\qquad x^4 + 10x^3 + 11x^2 + 8x + 12$

# Minimum distance of Reed-Solomon codes

**Proprerty**

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

# Minimum distance of Reed-Solomon codes

## Proprerty

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

## Démonstration.

Singleton bound : $\delta \leq n - k + 1$

$\square$

# Minimum distance of Reed-Solomon codes

## Proprerty

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

## Démonstration.

Singleton bound : $\delta \leq n - k + 1$
Let $w_f, w_g \in \mathcal{C}$, $\exists f, g \in K[X]_{<k}$ s.t. $w_f = (f(x_i))_i, w_g = (g(x_i))_i$.
If $f(x_i) \neq g(x_i)$ for $d < n - k + 1$ values $x_i$,
Then $f(x_j) - g(x_j) = 0$ for at least $n - d > k - 1$ values $x_j$.
Now $\deg(f - g) < k$, hence $f = g$. $\square$

# Minimum distance of Reed-Solomon codes

## Proprerty

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

## Démonstration.

Singleton bound : $\delta \leq n - k + 1$
Let $w_f, w_g \in \mathcal{C}$, $\exists f, g \in K[X]_{<k}$ s.t. $w_f = (f(x_i))_i, w_g = (g(x_i))_i$.
If $f(x_i) \neq g(x_i)$ for $d < n - k + 1$ values $x_i$,
Then $f(x_j) - g(x_j) = 0$ for at least $n - d > k - 1$ values $x_j$.
Now $\deg(f - g) < k$, hence $f = g$. □

$\Rightarrow$detect up to $n - k$ errors.
$\Rightarrow$correct up to $\frac{n-k}{2}$ errors.

# Decoding with the key equation

Let $P$ be the interpolant : $P(x_i) = y_i$ for all $1 \leq i \leq n$.

$$f(x_i) = P(x_i)$$

# Decoding with the key equation

Let $P$ be the interpolant : $P(x_i) = y_i$ for all $1 \leq i \leq n$.

⇒Equivalence evaluation/linear remainder

**Example**

$$\begin{aligned} P(X) \mod X-3 &= P(X) \text{ avec } (X-3=0) \\ &= P(X) \text{ avec } (X=3) \\ &= P(3) \end{aligned}$$

# Decoding with the key equation

Let $P$ be the interpolant : $P(x_i) = y_i$ for all $1 \leq i \leq n$.

$$f = P \mod \prod_{i=1}^{n}(x - x_i)$$

# Decoding with the key equation

Let $P$ be the erroneous interpolant : $P(x_i) = y_i + e_i$ for all $1 \leq i \leq n$.

$$f = P \mod \prod_{i \mid e_i = 0} (x - x_i)$$

# Decoding with the key equation

Let $P$ be the erroneous interpolant : $P(x_i) = y_i + e_i$ for all $1 \leq i \leq n$.

$$\Lambda f = \Lambda P \mod \prod_{i=1}^{n} (x - x_i)$$

and $\Lambda = \prod_{i | e_i \neq 0} (x - x_i)$

# Decoding with the key equation

Let $P$ be the erroneous interpolant : $P(x_i) = y_i + e_i$ for all $1 \leq i \leq n$.

$$N = \Lambda P \mod \prod_{i=1}^{n} (x - x_i)$$

and $\Lambda = \prod_{i | e_i \neq 0} (x - x_i)$ (Linearization)

# Berlekamp-Welch decoding

Find $N$ of degree $< k + t$ and $\Lambda$ of degree $\leq t$ such that

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

# Berlekamp-Welch decoding

Find $N$ of degree $< k+t$ and $\Lambda$ of degree $\leq t$ such that

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

## Linear system resolution

$N(X) = n_0 + \ldots n_{k+t-1}X^{k+t-1}$ et $\Lambda(X) = \ell_0 + \cdots + \ell_{t-1}X^{t-1} + X^t$.

Unknowns : $n_0, \ldots n_{k+t-1}, \ell_0, \ldots, \ell_{t-1}$ ($k+2t$ unknowns)

Equations : each evaluation in $x_i$ ($n$ equations)

$$\left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \ldots & x_1^{k+t-1} & \\ 1 & x_2 & x_1^2 & \ldots & x_1^{k+t-1} & \begin{bmatrix} -P(x_1) & & \\ & \ddots & \\ & & -P(x_n) \end{bmatrix} \begin{bmatrix} 1 & x_1 & \ldots & x_1^t \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^t \end{bmatrix} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 1 & x_n & x_n^2 & \ldots & x_n^{k+t-1} & \end{array}\right] \begin{bmatrix} n_0 \\ \vdots \\ n_{k+t-1} \\ \ell_0 \\ \ldots \\ \ell_{t-1} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

# Rational fraction reconstruction

**Problem (RFR : Rational Fraction Reconstruction)**

*Given* $A, B \in K[X]$ *with* $\deg B < \deg A = n$, *find* $f, g \in K[X]$, *with* $\deg f \leq d_F, \quad \deg g \leq n - d_F - 1$ *et*

$$f = gB \mod A.$$

# Rational fraction reconstruction

## Problem (RFR : Rational Fraction Reconstruction)

*Given* $A, B \in K[X]$ *with* $\deg B < \deg A = n$, *find* $f, g \in K[X]$, *with* $\deg f \leq d_F, \quad \deg g \leq n - d_F - 1$ *et*

$$f = gB \mod A.$$

$$\textit{i.e. } f(X) = g(X) \cdot B(X) + v(X) \cdot A(X)$$

# Rational fraction reconstruction

## Problem (RFR : Rational Fraction Reconstruction)

*Given* $A, B \in K[X]$ *with* $\deg B < \deg A = n$, *find* $f, g \in K[X]$, *with* $\deg f \leq d_F, \ \ \deg g \leq n - d_F - 1$ *et*

$$f = gB \mod A.$$

$$\textit{i.e.}\ f(X) = g(X) \cdot B(X) + v(X) \cdot A(X)$$

## Theorem

*Let* $(f_0 = A, f_1 = B, \ldots, f_\ell)$ *be the sequence of remainders in a run of the Extended Euclidean Algorithm applied to* $(A, B)$ *and* $u_i, v_i$ *the coefficients s.t.* $f_i = u_i f_0 + v_i f_1$*.*
*Then, at iteration* $j$ *s.t.* $\deg f_j \leq d_F < \deg f_{j-1}$*,*

1. $(f_j, v_j)$ *is a solution of the RFR problem.*
2. *it is minimal minimal : any other solution* $(f, g)$ *is of the form*

$$f = q f_j, \ \ g = q v_j \ \ \text{for } q \in K[X].$$

# Reed-Solomon decoding by the Extended Euclidean Algorithm

## Berlekamp-Welch by the Extended Euclidean Algorihtm

- Erroneous interpolant : $P = \text{Interp}((y_i, x_i))$
- Error locator polynomial : $\Lambda = \prod_{i|y_i \text{is erroneous}} (X - x_i)$

Find $f$ with $\deg f \leq d_F$ s.t. $f$ and $P$ coincide on $\geq n - t$ evaluations $x_i$.
$\Rightarrow$Extended Euclidean Alg. on $P$ and $\prod_{i=1}^{n} (X - x_i)$ :

$$\underbrace{\Lambda f}_{f_j} = \underbrace{\Lambda}_{g_j} P \mod \prod_{i=1}^{n} (X - x_i)$$

and $(f_j = \Lambda f, \Lambda = g_j)$ is minimal
$\Rightarrow$one only has to divide :

$$f(X) = f_j(X)/g_j(X).$$

# Codes cycliques

## Definition

Un code est cyclique s'il est stable par rotation circulaire :
$c = (c_0, \ldots c_{n-1}) \in \mathcal{C} \Rightarrow \rho(c) = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$

# Codes cycliques

## Definition

Un code est cyclique s'il est stable par rotation circulaire :
$c = (c_0, \dots c_{n-1}) \in \mathcal{C} \Rightarrow \rho(c) = (c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$

## Interprétation polynomiale

$$
\begin{aligned}
f_c(x) &= c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \\
f_{\rho(c)}(x) &= c_{n-1} + c_0 X + c_1 X^2 + \cdots + c_{n-2} X^{n-1} = X f_c(X) \mod X^n - 1
\end{aligned}
$$

# Codes cycliques

## Definition

Un code est cyclique s'il est stable par rotation circulaire :
$c = (c_0, \dots c_{n-1}) \in \mathcal{C} \Rightarrow \rho(c) = (c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$

## Interprétation polynomiale

$$\begin{aligned} f_c(x) &= c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \\ f_{\rho(c)}(x) &= c_{n-1} + c_0 X + c_1 X^2 + \cdots + c_{n-2} X^{n-1} = X f_c(X) \mod X^n - 1 \end{aligned}$$

Tout code cyclique

- est isomorphe à un sous ensemble de $\mathbb{F}_q[X]/(X^n - 1)$ stable par multiplication par $X$

# Codes cycliques

## Definition

Un code est cyclique s'il est stable par rotation circulaire :
$c = (c_0, \dots c_{n-1}) \in \mathcal{C} \Rightarrow \rho(c) = (c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$

## Interprétation polynomiale

$$
\begin{aligned}
f_c(x) &= c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \\
f_{\rho(c)}(x) &= c_{n-1} + c_0 X + c_1 X^2 + \cdots + c_{n-2} X^{n-1} = X f_c(X) \mod X^n - 1
\end{aligned}
$$

Tout code cyclique

- est isomorphe à un sous ensemble de $\mathbb{F}_q[X]/(X^n - 1)$ stable par multiplication par $X$
- est un ideal de $\mathbb{F}_q[X]/(X^n - 1)$

# Codes cycliques

## Definition

Un code est cyclique s'il est stable par rotation circulaire :
$c = (c_0, \dots c_{n-1}) \in \mathcal{C} \Rightarrow \rho(c) = (c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$

## Interprétation polynomiale

$$\begin{aligned} f_c(x) &= c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \\ f_{\rho(c)}(x) &= c_{n-1} + c_0 X + c_1 X^2 + \cdots + c_{n-2} X^{n-1} = X f_c(X) \mod X^n - 1 \end{aligned}$$

Tout code cyclique

- est isomorphe à un sous ensemble de $\mathbb{F}_q[X]/(X^n - 1)$ stable par multiplication par $X$
- est un ideal de $\mathbb{F}_q[X]/(X^n - 1)$
- est engendré par un diviseur de $X^n - 1$.

# Construction des codes cycliques

**Theorem**

*Soit* $g(X) = g_0 + g_1 X + \cdots + g_{n-k} X^{n-k}$ *un diviseur unitaire de* $X^n - 1$ *dans* $\mathbb{F}_q[X]$*.*
*Soit* $m = (g_0, \ldots, g_{n-k-1}, 1, 0, \ldots, 0) \in \mathbb{F}_q^n$*.*
*Alors* $(m, \rho(m), \ldots, \rho^k(m))$ *forme une base d'un code cyclique longueur* $n$ *et dimension* $k$*.*
*Tout code cyclique s'obtient par une telle construction.*

# Construction des codes cycliques

## Theorem

*Soit* $g(X) = g_0 + g_1 X + \cdots + g_{n-k} X^{n-k}$ *un diviseur unitaire de* $X^n - 1$ *dans* $\mathbb{F}_q[X]$.
*Soit* $m = (g_0, \ldots, g_{n-k-1}, 1, 0, \ldots, 0) \in \mathbb{F}_q^n$.
*Alors* $(m, \rho(m), \ldots, \rho^k(m))$ *forme une base d'un code cyclique longueur* $n$ *et dimension* $k$.
*Tout code cyclique s'obtient par une telle construction.*

## Interprétation polynomiale

- $\mathcal{C}$ et l'ensemble de combinaisons linéaires des $\rho^i(m)$.
- tout mot de $\mathcal{C}$ correspond à un $f(X) = \sum_{i=1}^{k} a_i X^i g(X)$.
- tout mot de $\mathcal{C}$ correspond à un $f(X) = g(X)h(X)$ pour $h \in \mathbb{F}_q[X]_{\leq k}$.

Détection d'erreur : $f \in \mathcal{C}$ ssi $g$ divise $f$.

# Construction des codes cycliques

Comment trouver un *bon* diviseur de $X^n - 1$ ?

$$X^n - 1 = \prod_{i=0}^{n-1} (X - \alpha^i)$$

pour $\alpha$ une racine primitive $n$ème de l'unité.

# Construction des codes cycliques

Comment trouver un *bon* diviseur de $X^n - 1$ ?

$$X^n - 1 = \prod_{i=0}^{n-1} (X - \alpha^i)$$

pour $\alpha$ une racine primitive $n$ème de l'unité.

**Cas facile : $\alpha \in \mathbb{F}_q$**

Tout sous-ensemble de $\alpha^i$ convient

# Construction des codes cycliques

Comment trouver un *bon* diviseur de $X^n - 1$ ?

$$X^n - 1 = \prod_{i=0}^{n-1} (X - \alpha^i)$$

pour $\alpha$ une racine primitive $n$ème de l'unité.

**Cas facile : $\alpha \in \mathbb{F}_q$**

Tout sous-ensemble de $\alpha^i$ convient

**Cas difficile : $\alpha \notin \mathbb{F}_q$**

Classes cyclotomiques :
$\prod_{i \in \Sigma} (X - \alpha^i) \in \mathbb{F}_q[X]$ ssi $\Sigma$ est stable par mult. par $q$ modulo $n$.

- Ainsi, si on prend $\alpha^j$, on doit prendre tous les éléments de $\Sigma_j = \{jq^k \mod n, k \in \mathbb{Z}\}$
- On prend donc des unions $\bigcup_j \Sigma_j$

# Construction des codes cycliques

**Theorem (BCH)**

*Si* $\Sigma \subset \{0, \ldots, n-1\}$ *contient* $s$ *entiers consécutifs, alors*

$$g(x) = \prod_{i \in \Sigma} (X - \alpha^i)$$

*génère un code cyclique de distance minimale* $\geq s + 1$*.*

## Example

$q = 2, n = 7$. On prend $\alpha$ une racine primitive 7ème de l'unité dans $(\mathbb{F}_8)^*$. $\Sigma_0 = \{0\}$, $\Sigma_1 = \{1, 2, 4\} = \Sigma_2 = \Sigma_4$, $\Sigma_3 = \{3, 6, 12 = 5 \mod 7\} = \Sigma_5 = \Sigma_6$

$$X^7 - 1 = (X - 1)\underbrace{(X - \alpha)(X - \alpha^2)(X - \alpha^4)}_{1+X+X^3}\underbrace{(X - \alpha^3)(X - \alpha^6)(X - \alpha^5)}_{1+X^2+X^3}$$

Pour $g(X) = 1 + X + X^3$, la matrice génératrice est

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

équivalente au code de Hamming $(7, 4, 3)$.

# Codes de Reed-Solomon cycliques

Cas particulier : $n = q - 1$

- $(\mathbb{F}_q)^*$ est cyclique, donc il existe $\alpha$ générateur, racine primitive $n$ème de l'unité dans $\mathbb{F}_q$.

# Codes de Reed-Solomon cycliques

Cas particulier : $n = q - 1$

- $(\mathbb{F}_q)^*$ est cyclique, donc il existe $\alpha$ générateur, racine primitive $n$ème de l'unité dans $\mathbb{F}_q$.
- Cas facile : on peut choisir les racines de $g$ arbitrairement.

# Codes de Reed-Solomon cycliques

Cas particulier : $n = q - 1$

- $(\mathbb{F}_q)^*$ est cyclique, donc il existe $\alpha$ générateur, racine primitive $n$ème de l'unité dans $\mathbb{F}_q$.
- Cas facile : on peut choisir les racines de $g$ arbitrairement.
- $g(X) = \prod_{i=t}^{t+d-1}(X - \alpha^i) \in \mathbb{F}_q[X]$ garantit une distance minimale de $d + 1$.

# Codes de Reed-Solomon cycliques

Cas particulier : $n = q - 1$

- $(\mathbb{F}_q)^*$ est cyclique, donc il existe $\alpha$ générateur, racine primitive $n$ème de l'unité dans $\mathbb{F}_q$.
- Cas facile : on peut choisir les racines de $g$ arbitrairement.
- $g(X) = \prod_{i=t}^{t+d-1}(X - \alpha^i) \in \mathbb{F}_q[X]$ garantit une distance minimale de $d + 1$.
- maximise la dimension pour une distance et une longueur donnée (MDS).

# Décodage en liste de Sudan

**Problem**

*Trouver la liste de tous les mots de code qui correspondent en au moins $t$ points.*

# Décodage en liste de Sudan

**Problem**

*Trouver la liste de tous les mots de code qui correspondent en au moins $t$ points.*

**Theorem (Sudan'97)**

*On peut calculer en temps polynomial cette liste de taille $\sqrt{\frac{n}{k}}$ pour $t > \sqrt{2kn}$.*

# Décodage en liste de Sudan

**Problem**

*Trouver la liste de tous les mots de code qui correspondent en au moins $t$ points.*

**Theorem (Sudan'97)**

*On peut calculer en temps polynomial cette liste de taille $\sqrt{\frac{n}{k}}$ pour $t > \sqrt{2kn}$.*

Généralise Berlekamp-Welsh : $Q(X,Y) = \underbrace{N(X)}_{\Lambda(X)f(X)} - Y\Lambda(X) = 0$ avec $\deg_Y > 1$

1. Trouver $Q(X,Y) \in \mathbb{F}[X,Y]$ t.q. $Q(x_i, y_i) = 0$ pour tout $i$
2. Trouver ses facteurs linéaires en $Y$ : $Y - f(X)$
3. Retourner la liste de ces $f(X)$ t.q. $f(x_i) = y_i$ en au moins $t$ points

# Power decoding

Idée : générer $\ell$ relation indépendantes en élevant à la puissance.

$$\begin{cases} \Lambda(x_i)f(x_i) & = & y_i\Lambda(x_i) \\ \Lambda(x_i)f(x_i)^2 & = & y_i^2\Lambda(x_i) \\ \vdots & \vdots & \vdots \\ \Lambda(x_i)f(x_i)^\ell & = & y_i^\ell\Lambda(x_i) \end{cases}$$

# Power decoding

Idée : générer $\ell$ relation indépendantes en élevant à la puissance.

$$\left\{\begin{array}{ccc} \Lambda(x_i)f(x_i) & = & y_i\Lambda(x_i) \\ \Lambda(x_i)f(x_i)^2 & = & y_i^2\Lambda(x_i) \\ \vdots & \vdots & \vdots \\ \Lambda(x_i)f(x_i)^\ell & = & y_i^\ell\Lambda(x_i) \end{array}\right.$$

- $e + \sum_{i=1}^{\ell} ik$ inconnues
- $n\ell$ equations

Solution unique (si généricité) si $\frac{\ell(\ell+1)}{2}k + e = n\ell$.
Pour $\ell \approx \sqrt{k/n}$ et $n - e \approx \sqrt{2kn}$.

# Power decoding

Idée : générer $\ell$ relation indépendantes en élevant à la puissance.

$$\left\{\begin{array}{rcl} \Lambda(x_i)f(x_i) & = & y_i\Lambda(x_i) \\ \Lambda(x_i)f(x_i)^2 & = & y_i^2\Lambda(x_i) \\ \vdots & \vdots & \vdots \\ \Lambda(x_i)f(x_i)^\ell & = & y_i^\ell\Lambda(x_i) \end{array}\right.$$

- $e + \sum_{i=1}^{\ell} ik$ inconnues
- $n\ell$ equations

Solution unique (si généricité) si $\frac{\ell(\ell+1)}{2}k + e = n\ell$.
Pour $\ell \approx \sqrt{k/n}$ et $n - e \approx \sqrt{2kn}$.

- capacité de correction identique à Sudan.
- mais pas de liste
- peut échouer