Cryptosystèmes basés sur les réseaux et prouvés sûrs

Guillaume Dabosville

dabosvil@info.unicaen.fr

GREYC - Equipe Informatique - Université de Caen

Page de pub...

`http://www.info.unicaen.fr/~dabosvil/CAEN2003/`

Plan de l'exposé

- Prouver un cryptosystème ;
  - Rappels sur la théorie de la complexité ;
  - Preuve de sécurité ;
- Connection *cas le pire/cas moyen* d'Ajtai ;
  - Réseaux euclidiens et problèmes algorithmiques ;
  - La classe de réseaux d'Ajtai ;
  - La connection ;
- Un cryptosystème “prouvé sûr” ;

Preuves de sécurité

BUT

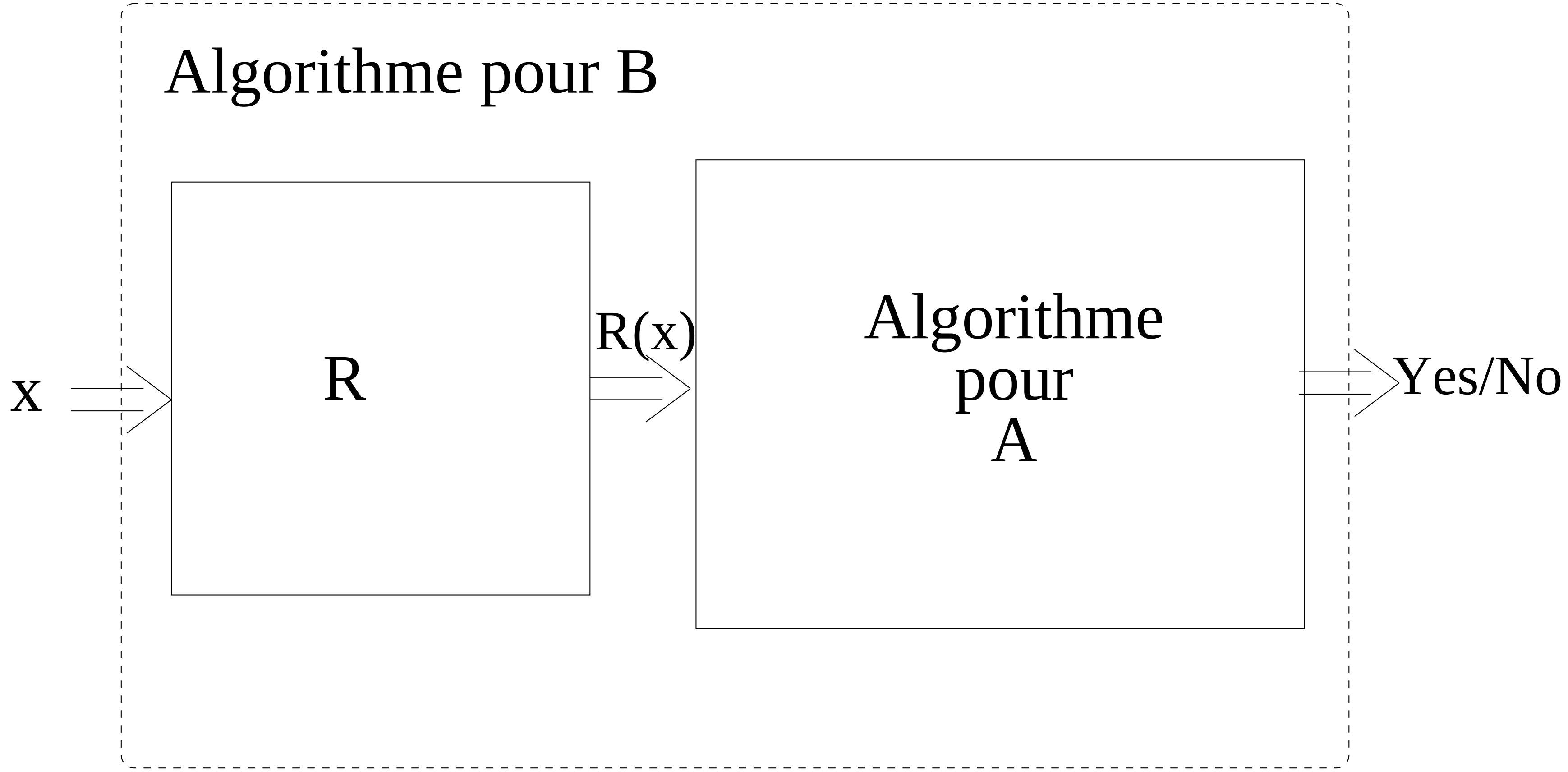
Quantifier la difficulté de “casser” ou “résoudre” un cryptosystème

MÉTHODE

Utiliser la théorie de la complexité (classes de complexité)

Preuves de sécurité

Réduction de B à A



Preuves de sécurité

Un problème A est NP-complet si

$A \in NP$ (appartenance)

$\forall B \in NP$, B se réduit à A (NP-dur)

Preuves de sécurité

$\mathcal{A}$ est “difficile” peut signifier :

$\mathcal{A}$ est NP-complet

$\mathcal{A}$ est NP-dur

$\mathcal{A}$ n’est pas difficile au sens “théorie de la complexité”

mais “difficile” par expérience (FACTORISATION)

Preuves de sécurité

Soit $\mathcal{C}$ un cryptosystème

"Prouver" $\mathcal{C}$

$=$

Trouver un problème $\mathcal{P}$ (*réputé*) difficile

et

réduire $\mathcal{P}$ à $\mathcal{C}$

Preuves sécurité - Raisonnement

Hyp : $\mathcal{P}$ un problème réputé difficile ;

$\exists$ une attaque A qui “casse” $\mathcal{C}$

Si $\exists\, R : \mathcal{P} \to \mathcal{C}$

Alors $\mathcal{P}$ est résolu

contradiction !

Preuves de sécurité - Limites

$\exists$ des instances difficiles

Quelle distribution ?

Facile à générer ?

Connection d'Ajtai - Idée

IDÉE

Intégrer la notion de difficulté en moyenne

MÉTHODE

Relier le pire des cas d'un problème A au cas moyen d'un problème B

A et B sont des problèmes dans les réseaux

Définition d'un réseau de $\mathbb{R}^n$

$b_1, \cdots, b_p$ $(p \leq n)$ des vecteurs de $\mathbb{R}^n$ linéairement indépendants

$L(b_1, \cdots, b_p)$ l'ensemble des combinaisons linéaires à coefficients entiers de $b_1, \cdots, b_p$ est un réseau de $\mathbb{R}^n$.

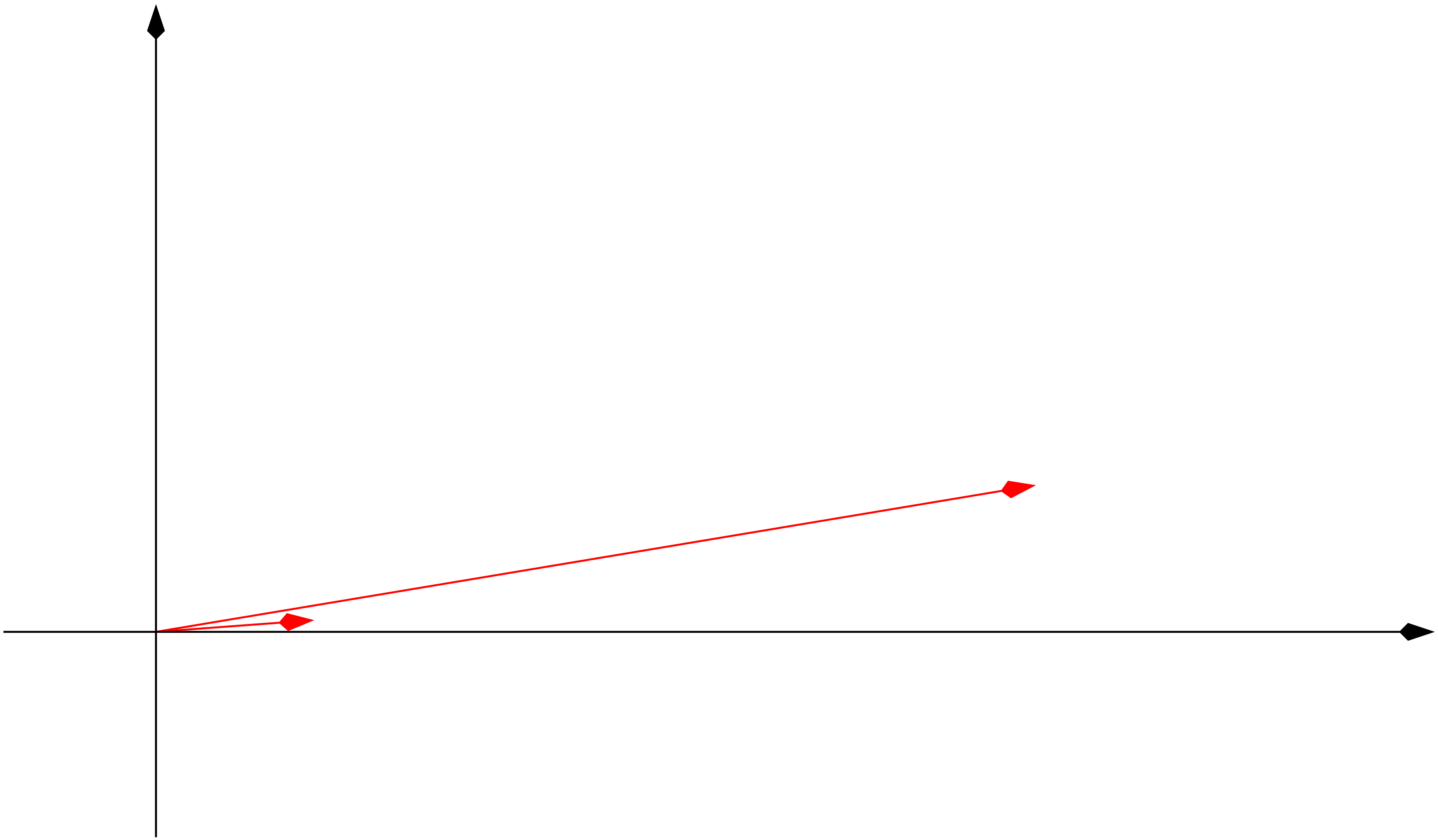
Bases et dimension d'un réseau

Soit $L = L(b_1, \cdots, b_p)$ un réseau.

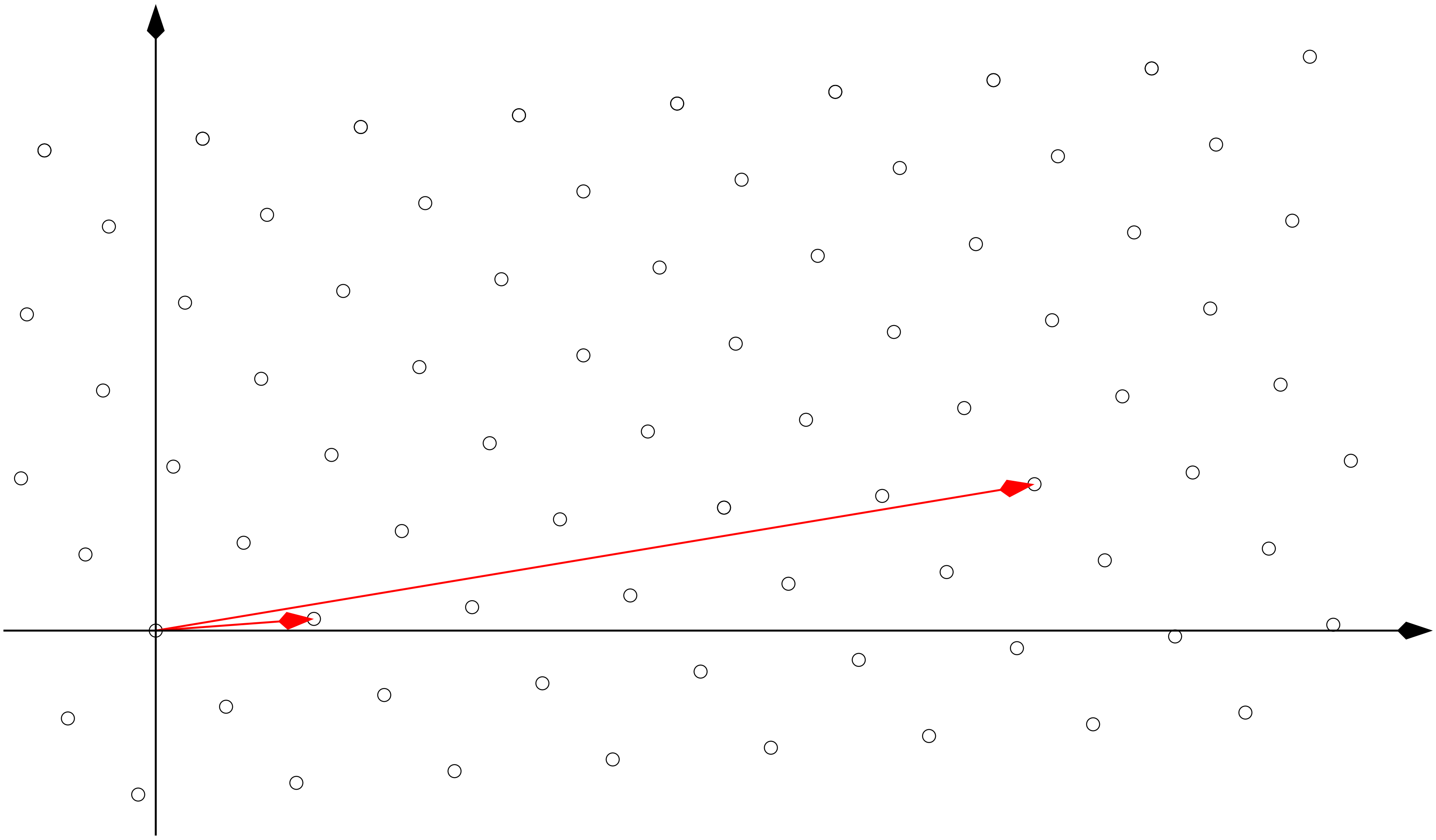
- $b_1, \cdots, b_p$ est une base de L ;
- Il en existe une infinité ;
- Dimension de L : cardinal d'une base de L (ie p) ;

un exemple...

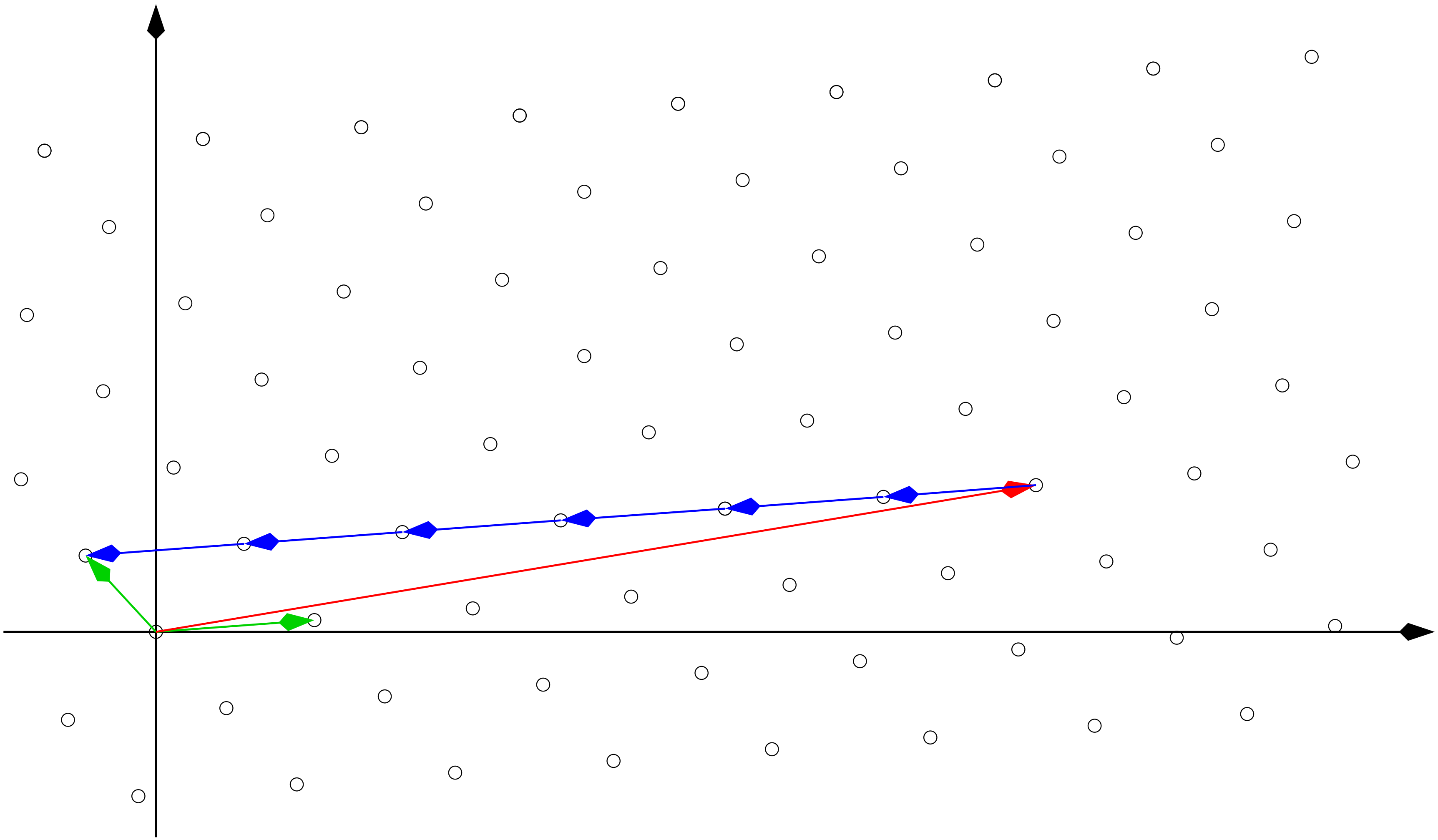
Réseau en dimension 2



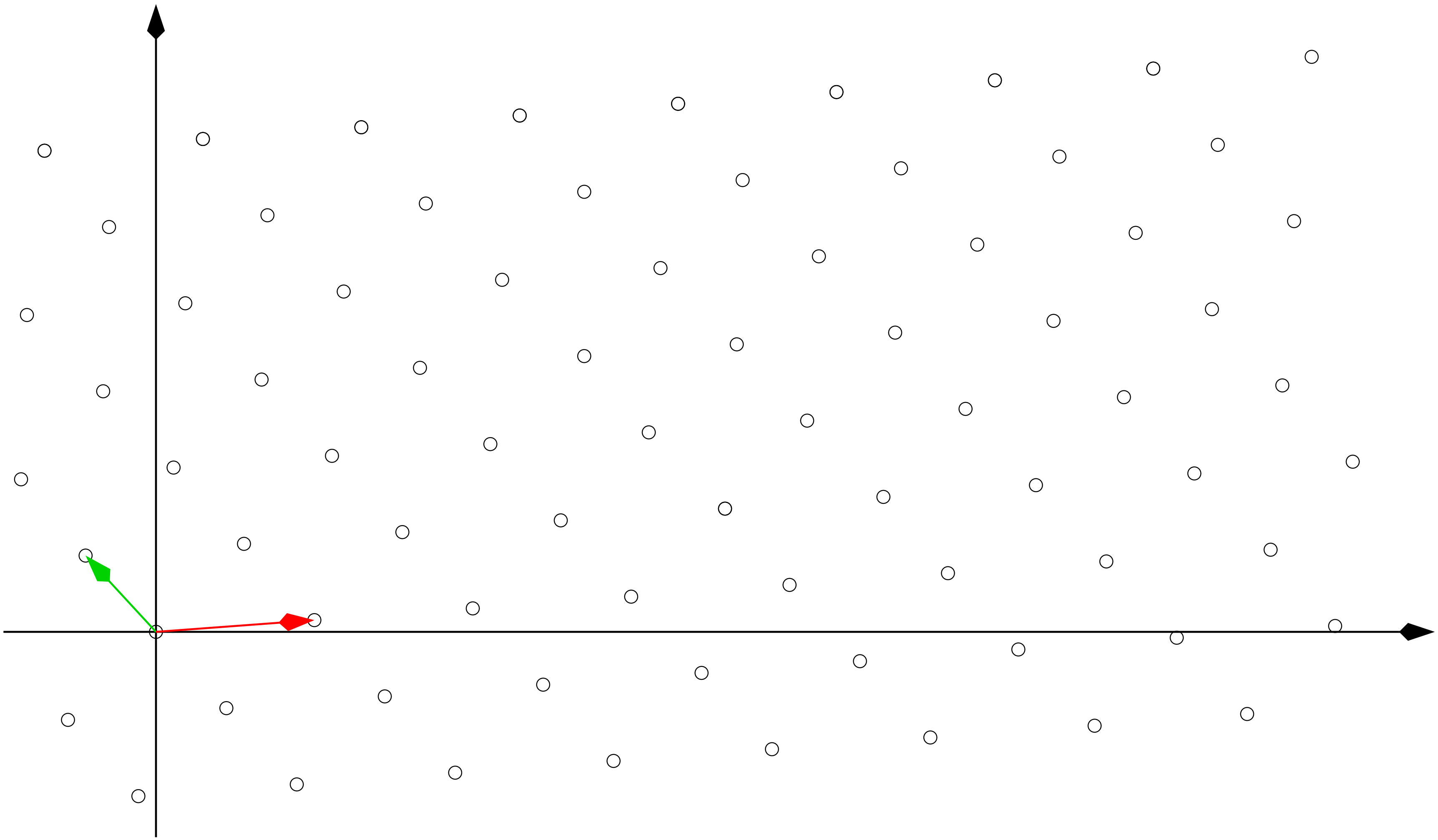
Réseau en dimension 2



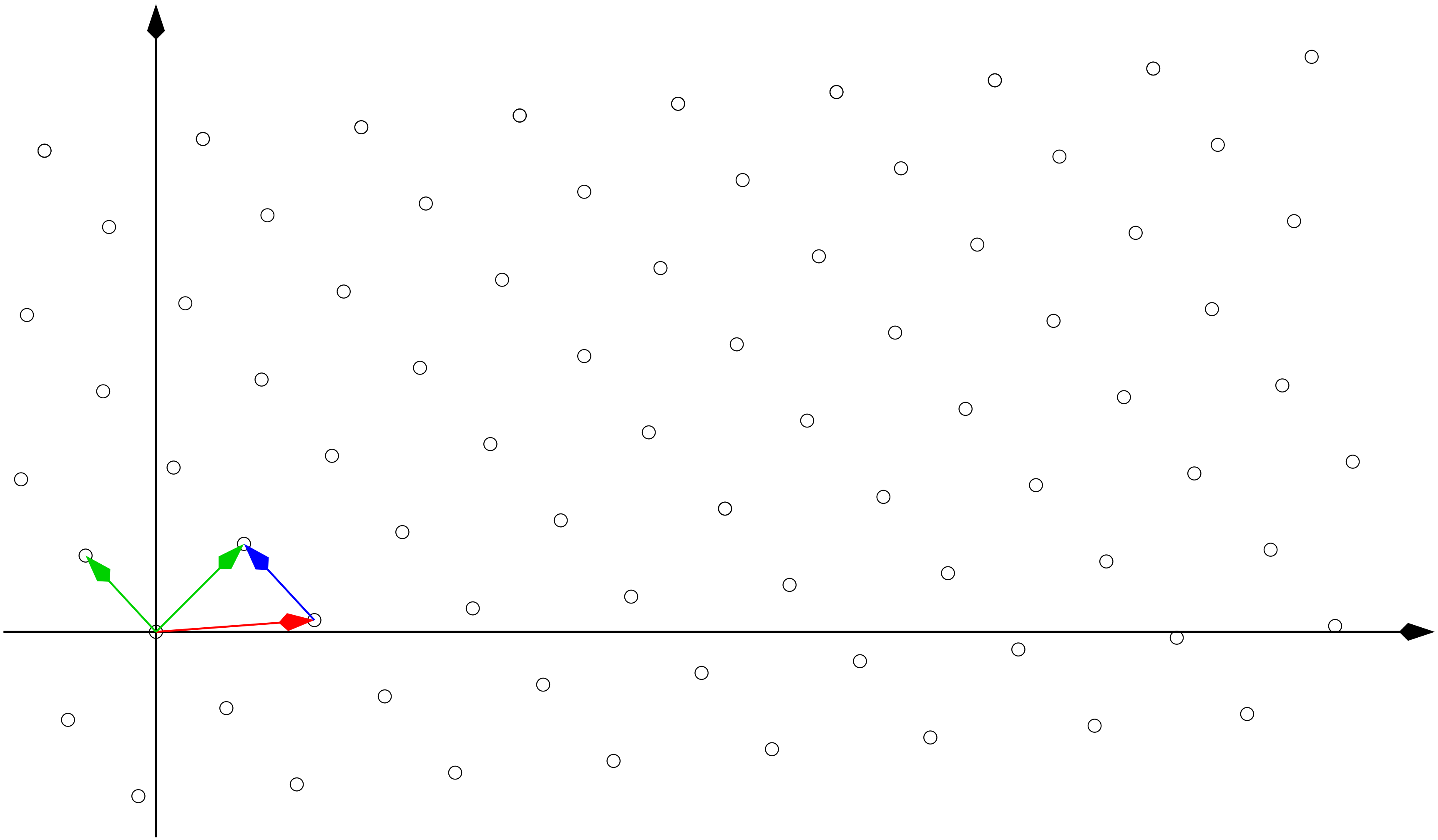
Réseau en dimension 2



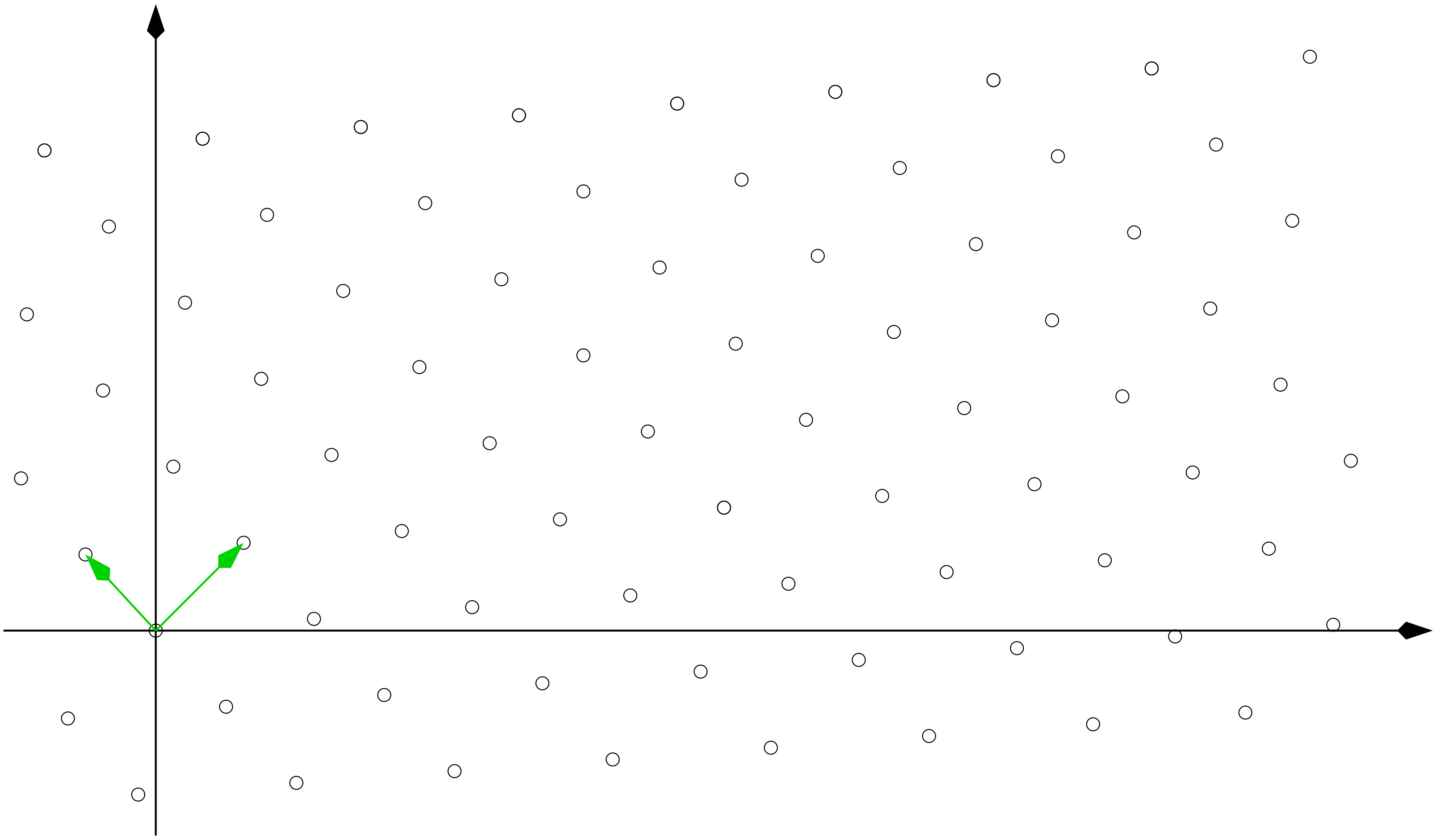
Réseau en dimension 2



Réseau en dimension 2



Réseau en dimension 2



Invariants d'un réseau

Le déterminant

$L = L(b_1, \cdots, b_p)$ un réseau de dimension p ;

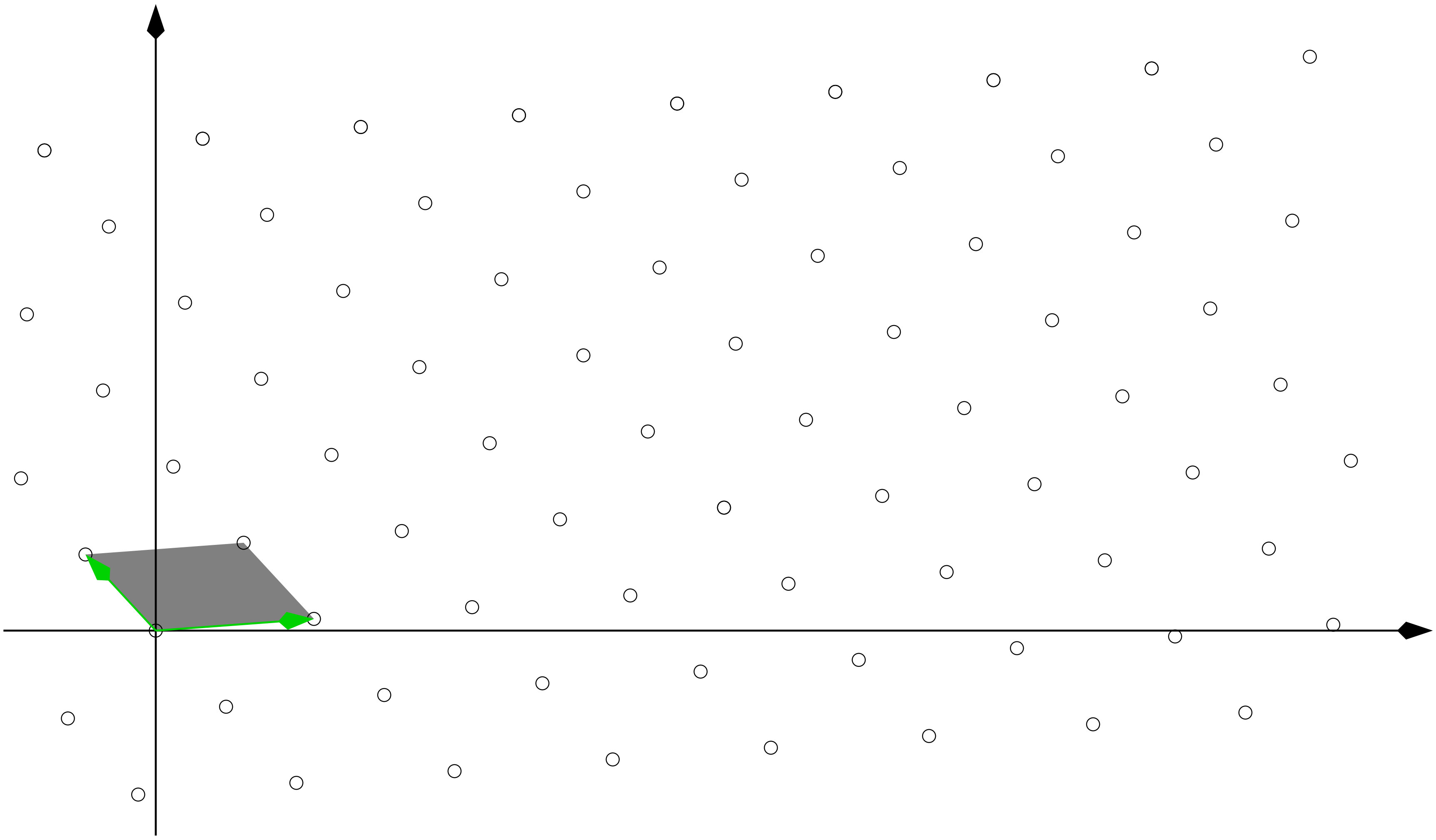
$B = (b_{ij})_{i,j}$ matrice représentant L ;

$G = B^T.B$ matrice de Gram ;

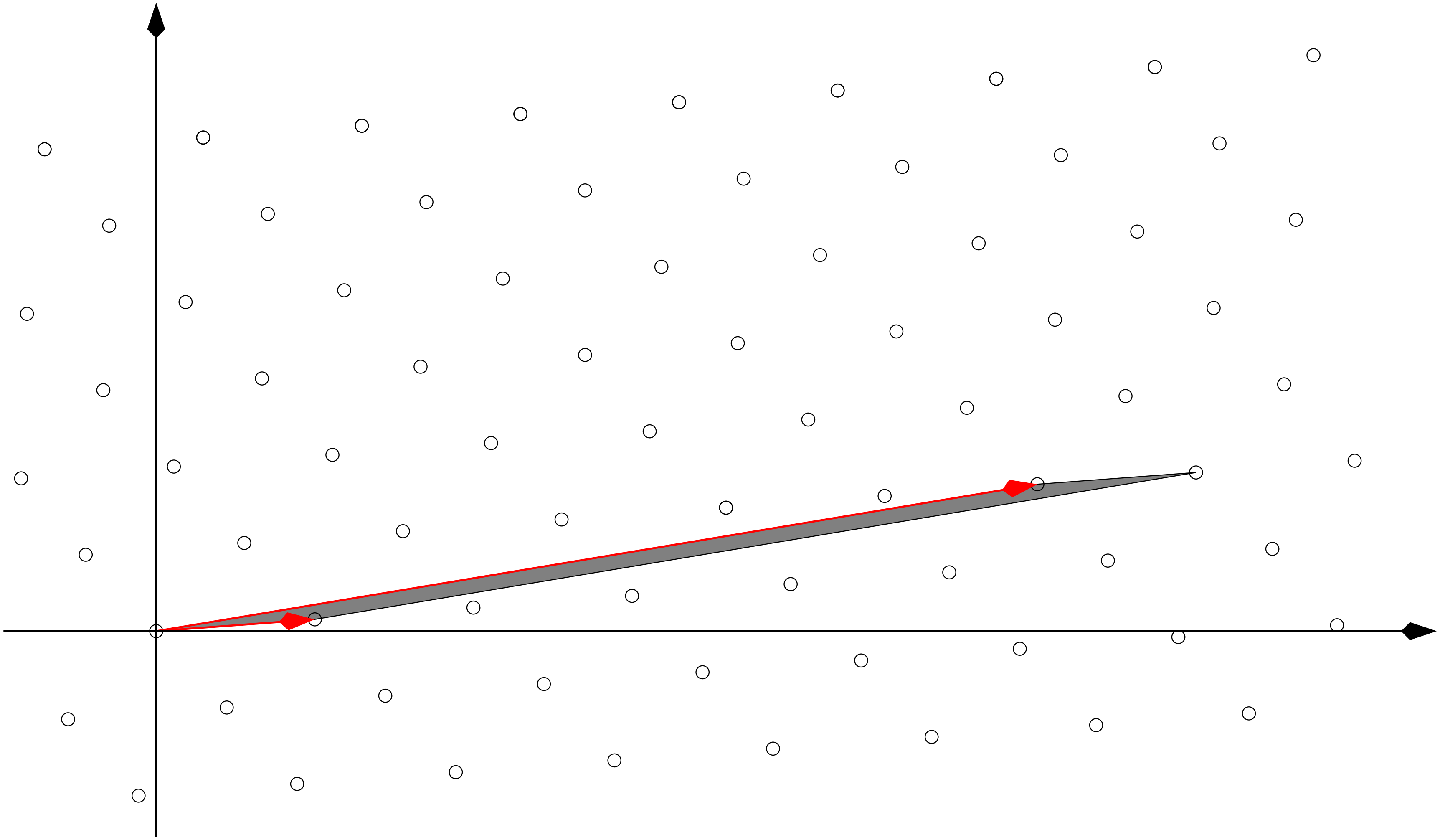
$\det(L) = \sqrt{\det(G)}$ le déterminant de L

est un invariant.

Interprétation géométrique



Interprétation géométrique



Invariants d'un réseau

Les minima successifs

$\lambda_1(L)$: le plus petit $r > 0$ tel que
$B(0, r) \cap (L - \{0\}) \neq \emptyset$;

Invariants d'un réseau

Les minima successifs

$\lambda_i(L)$: le plus petit $r > 0$ tel que $B(0, r)$ contienne i vecteurs de L linéairement indépendants ;

$\mathrm{bl}_i(L)$: le plus petit $r > 0$ tel que $B(0, r)$ contienne i vecteurs de L pouvant être complétés en une base de L ;

$$\lambda_n(L) \leq \mathrm{bl}_n(L) \quad ;$$

Problèmes algorithmiques

SVP : retrouver $\lambda_1(L)$;

CVP : étant donné $x \in \mathbb{R}^n$ trouver $y \in L$ tel que dist(x, y) est minimale ;

SB : $\mathrm{bl}_n(L)$;

Autres problèmes Algorithmiques

Approximer les problèmes SVP, CVP et SB ;

Trouver le plus court vecteur d'un réseau possédant un n^c-unique plus court vecteur ;

Approximer SVP

Soit L un réseau de dimension n.

Trouver $v \in L$ t.q.

$$\frac{\lambda_1(L)}{n^{c_2}} \leq \|v\| \leq n^{c_1}\lambda_1(L)$$

v est une approximation de $\lambda_1(L)$ à un facteur polynomial près.

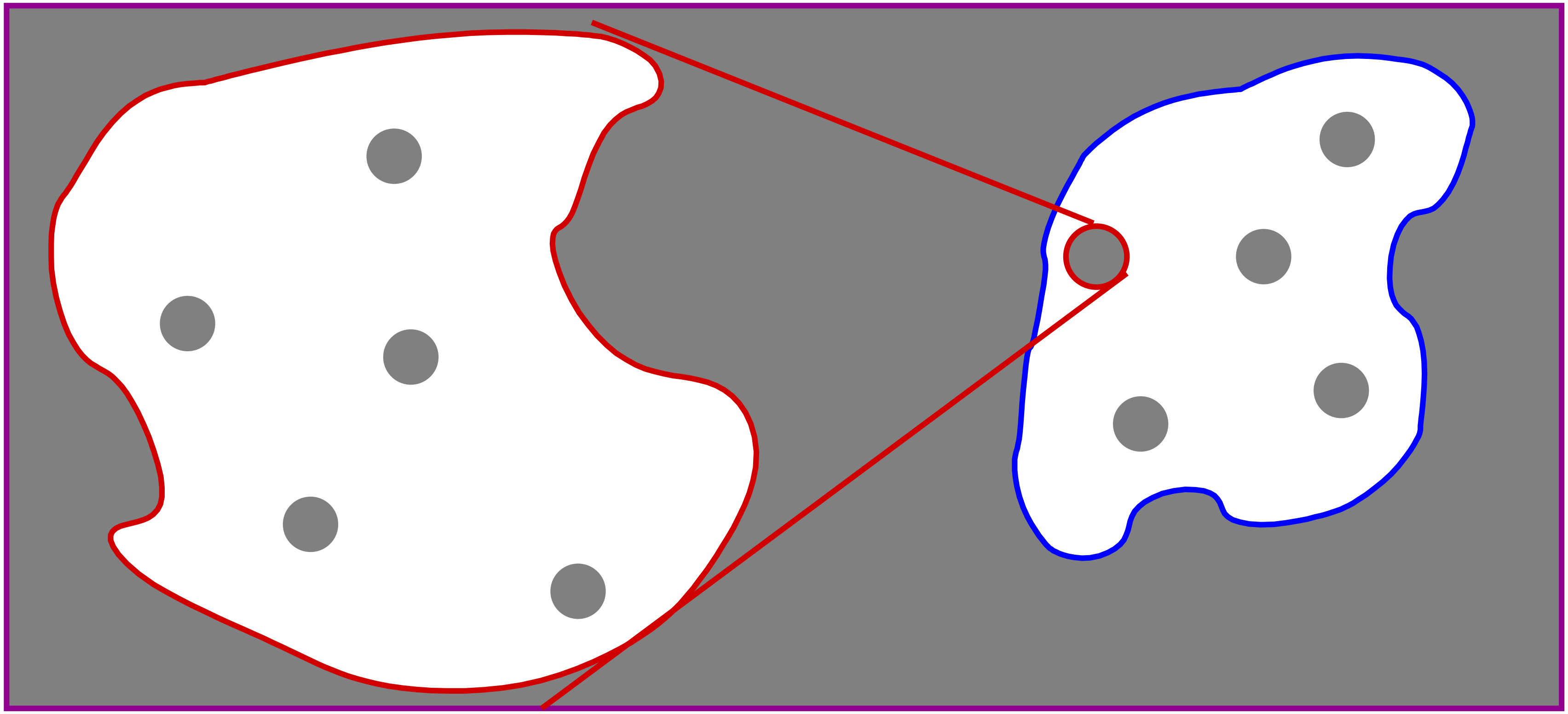
Le n^c-unique plus court vecteur

L un réseau de dimension n contient un n^c-unique plus court vecteur v si

$$\forall u \in L \ : \ (\|u\| \leq n^c \|v\|) \Rightarrow u // v$$

Connection d'Ajtai - Idée

Problème A Problème B



Connection d'Ajtai

Problème $\mathcal{A}$

$$=$$

Approximer SVP

Approximer SB

Retrouver le n^c-unique plus court vecteur

Les réseaux d'Ajtai

- $\lambda = < u_1, \cdots, u_m >$ où $u_i \in_R \mathbb{Z}^n$ $(m = n \log n)$;
- $q \in \mathbb{Z}$

$\Gamma_{\lambda,q} = \{h = (h_1, \cdots, h_m),\ h_i \in \mathbb{Z} \mid \sum_{i=1}^{m} h_i u_i \equiv 0 \pmod q\}$ est un réseau de dimension m

Connection d'Ajtai

$$\text{Problème } \mathcal{B}$$
$$=$$
$$\text{Trouver } v \text{ t.q. } \|v\| \leq n \text{ dans } \Gamma_{\lambda,q}$$

La connection - Théorème

Si algo A trouve $v \in \Gamma_{\lambda,q}$ t.q. $\|v\| \leq n$ avec une proba. $> \frac{1}{2}$

alors $\exists$ algo B qui vérifie :

Entrée : $a_1, \cdots, a_n \in \mathbb{Z}^n$

Sortie : $z, u, < d_1, \cdots, d_n >$

Algo B

$L = L(a_1, \cdots, a_n)$

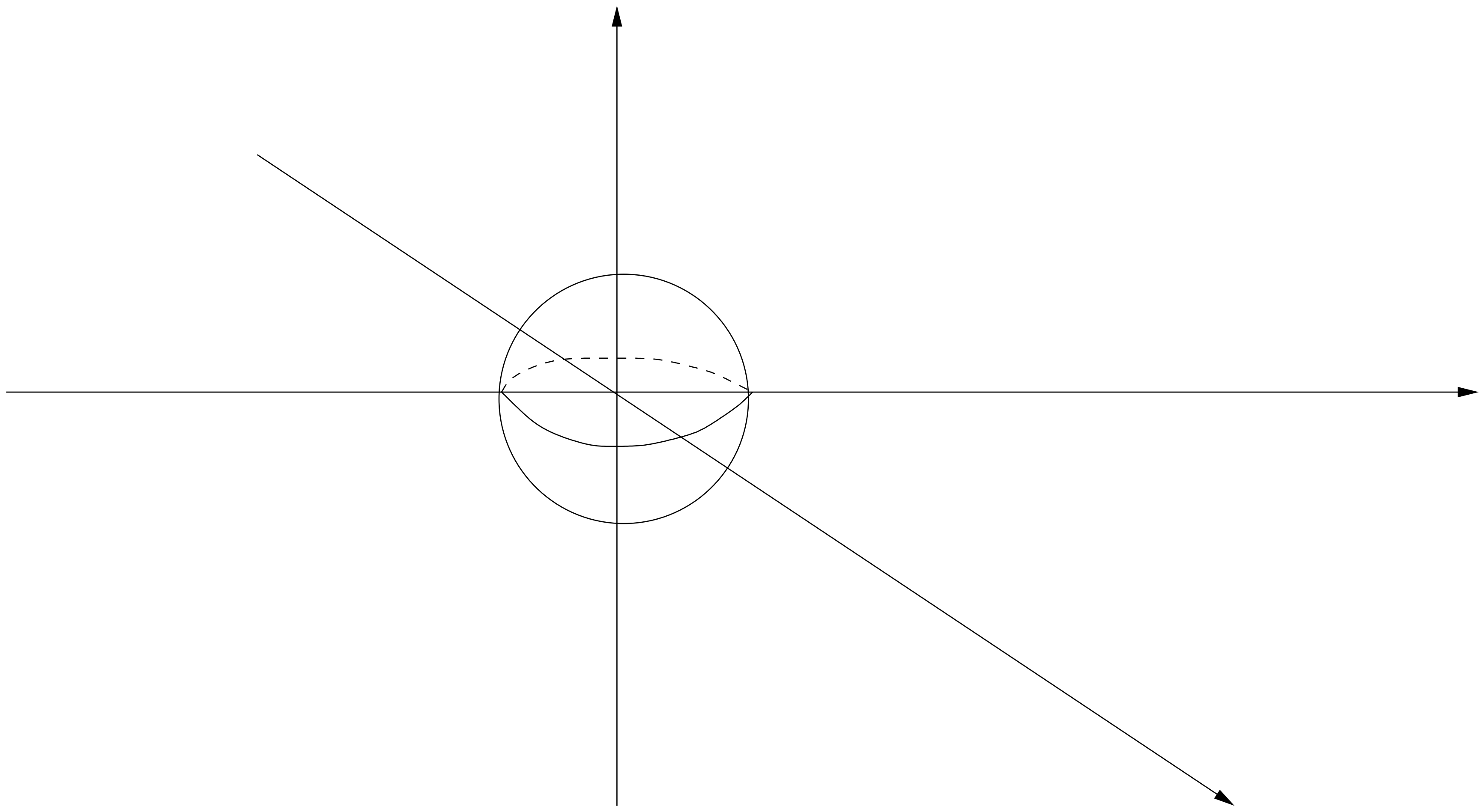
1. $$\frac{\lambda_1(L)}{n^c} \leq z \leq \lambda_1(L)$$
2. si $\lambda_1(L)$ est un n^c-unique plus court vecteur de L alors $u = \lambda_1(L)$
3. $d_1, \cdots, d_n$ est une base de L avec $\max_{i=1}^{n} \|d_i\| \leq n^{c'} \mathsf{bl}_n(L)$

Un cryptosystème “prouvé sûr”

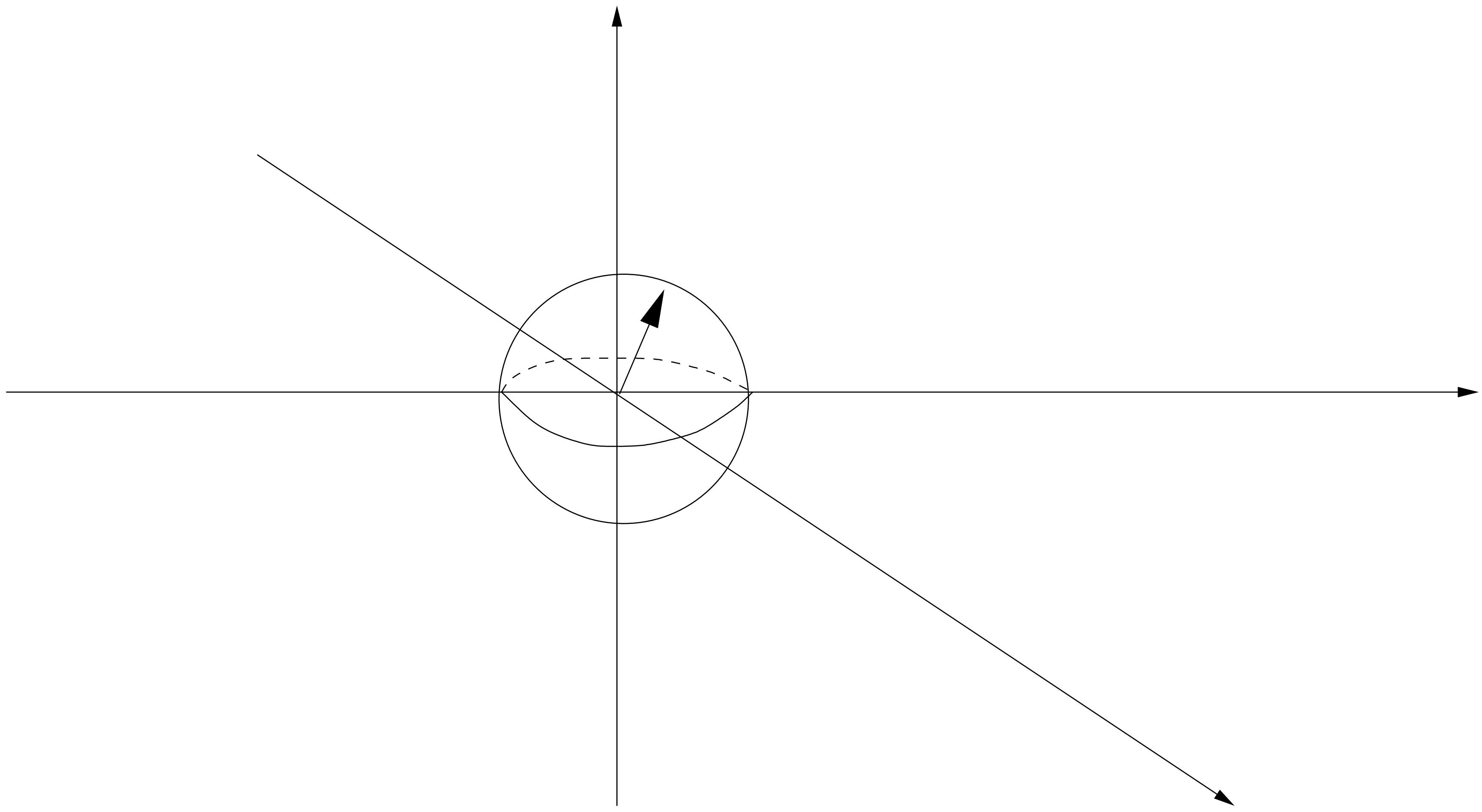
Génération des clés

Choisir u aléatoirement dans $B(0,1)$;

Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”



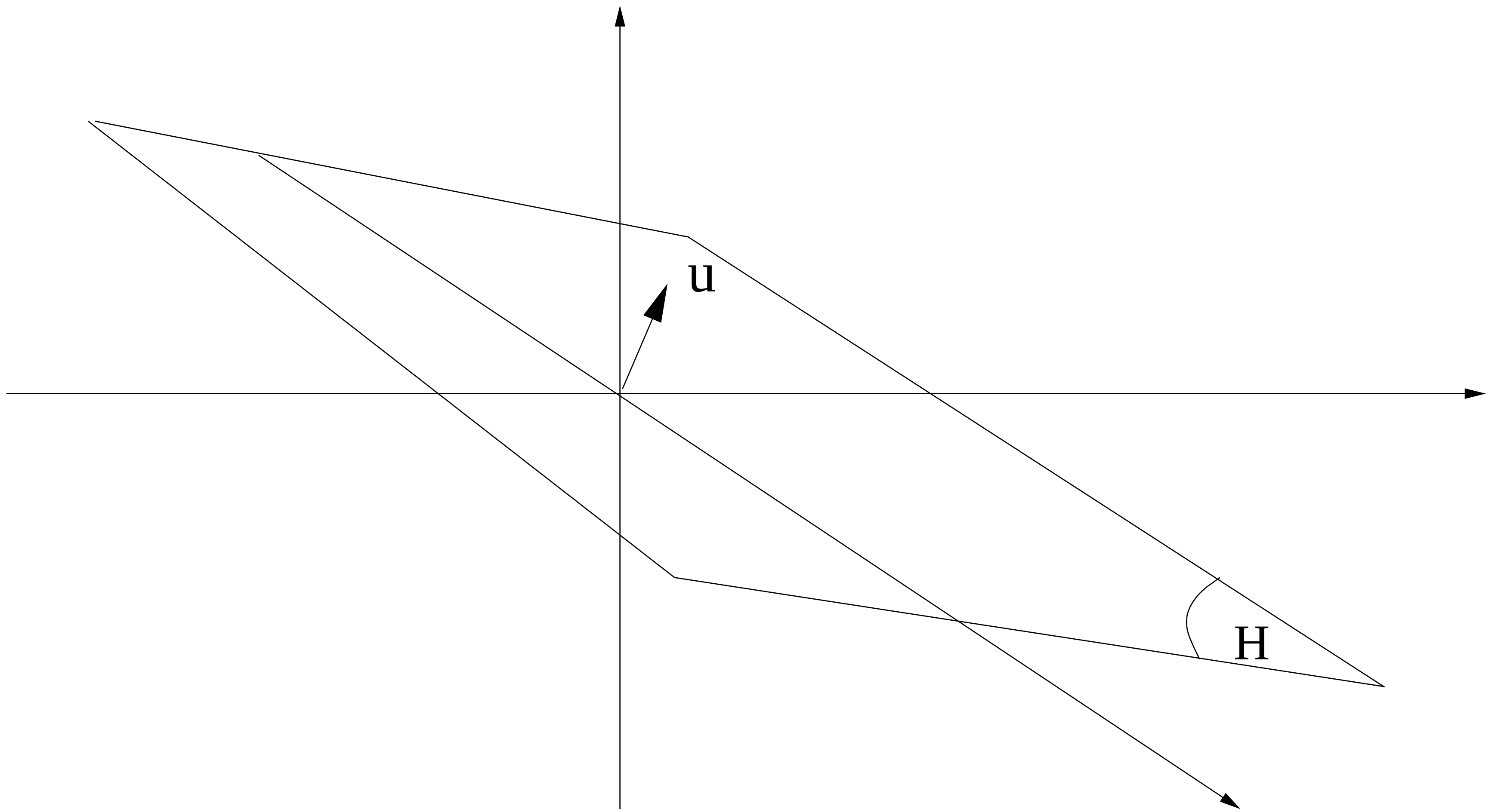
Un cryptosystème “prouvé sûr”

Génération des clés

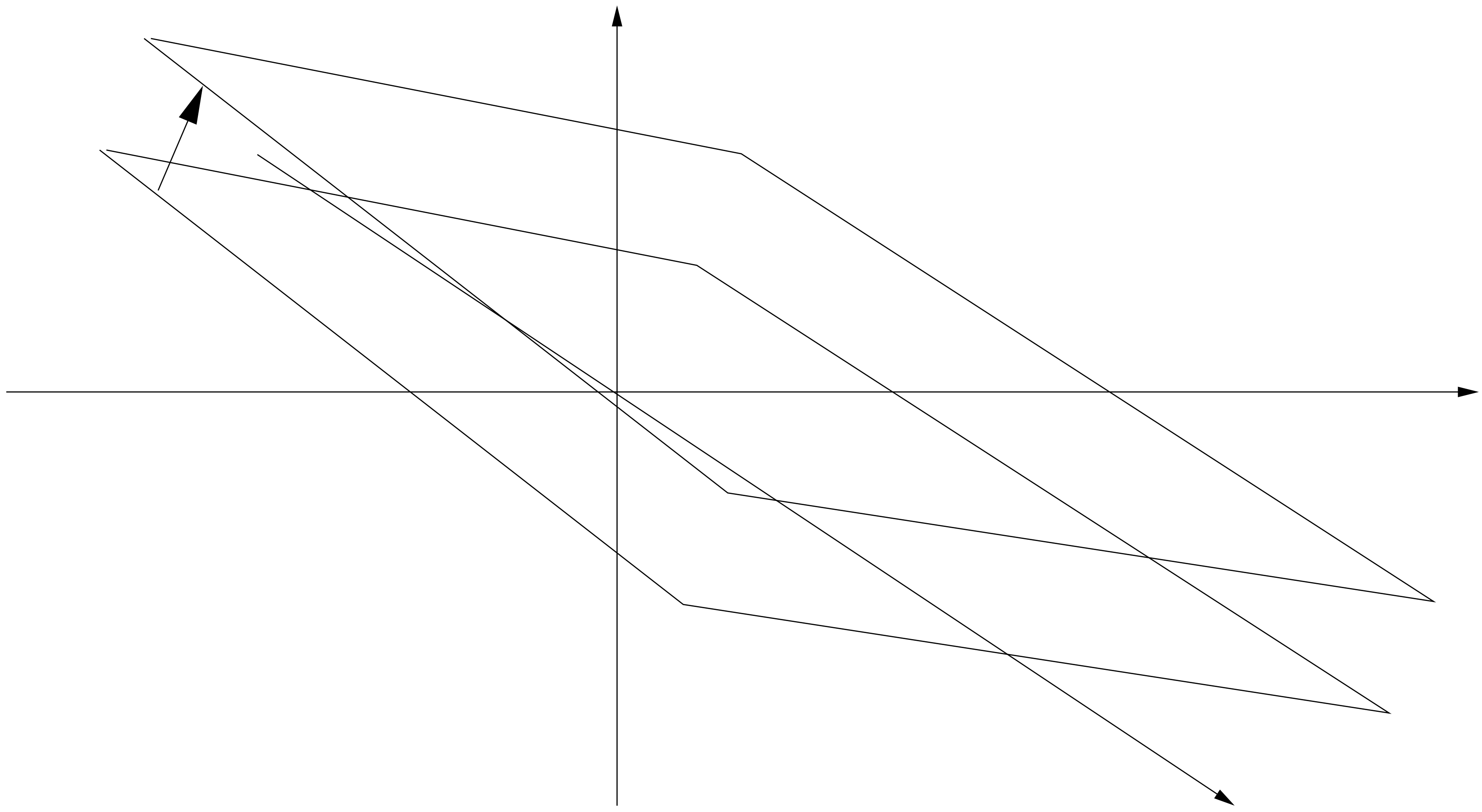
Choisir u aléatoirement dans $B(0,1)$;

Construire $X = \{x \in \mathbb{R}^n \mid <x, u> \in \mathbb{Z}\}$;

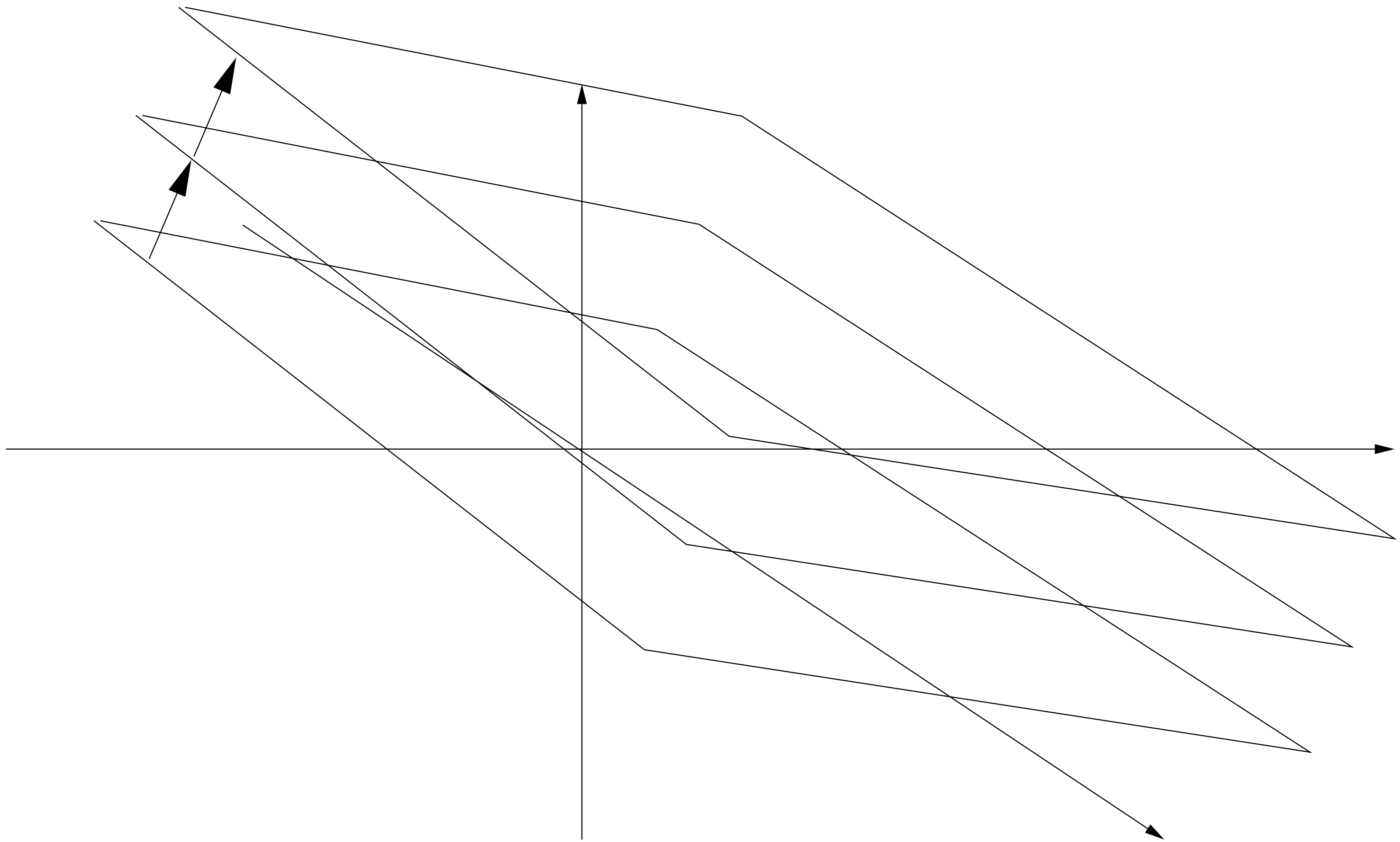
Un cryptosystème "prouvé sûr"



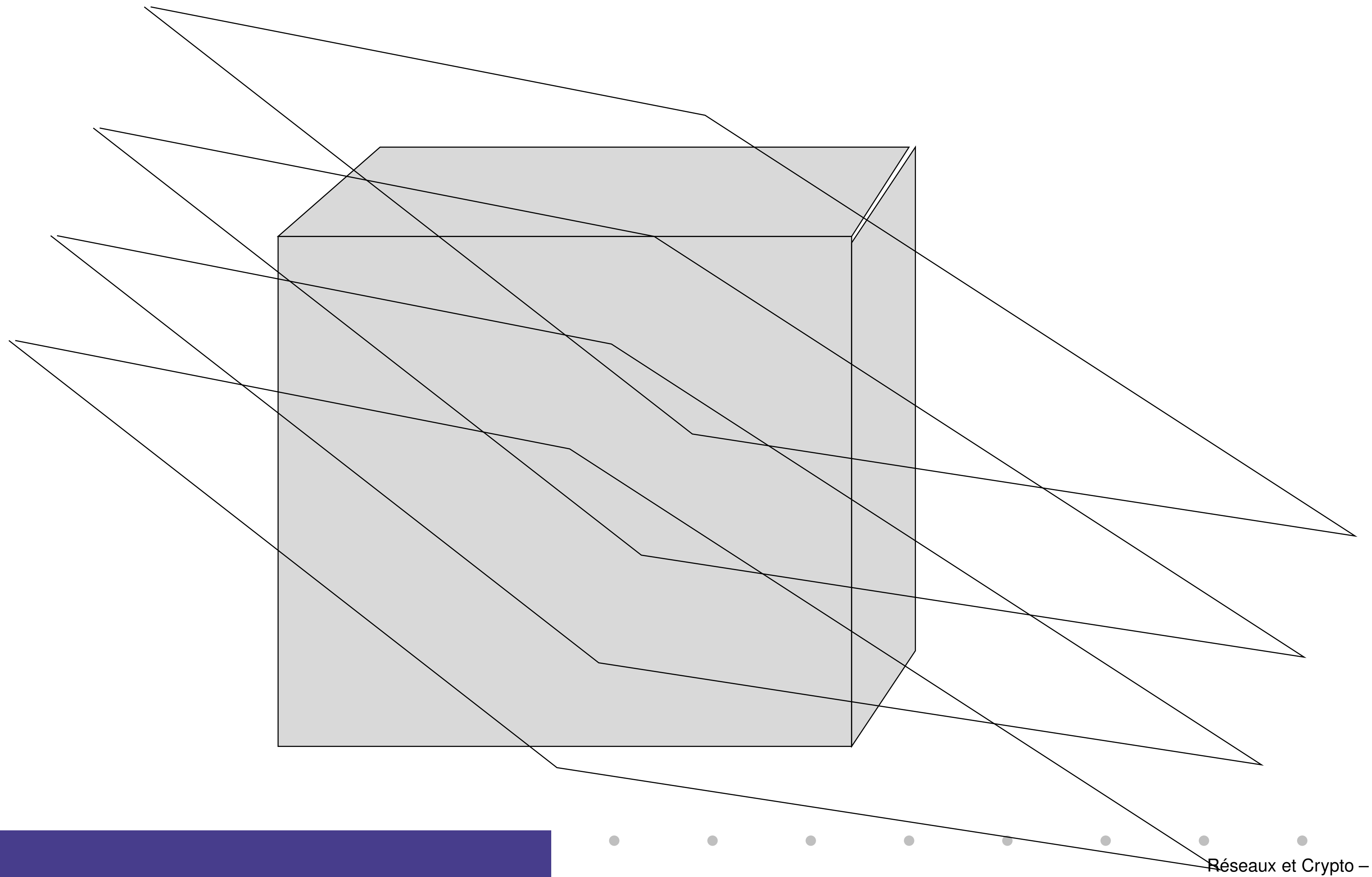
Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”

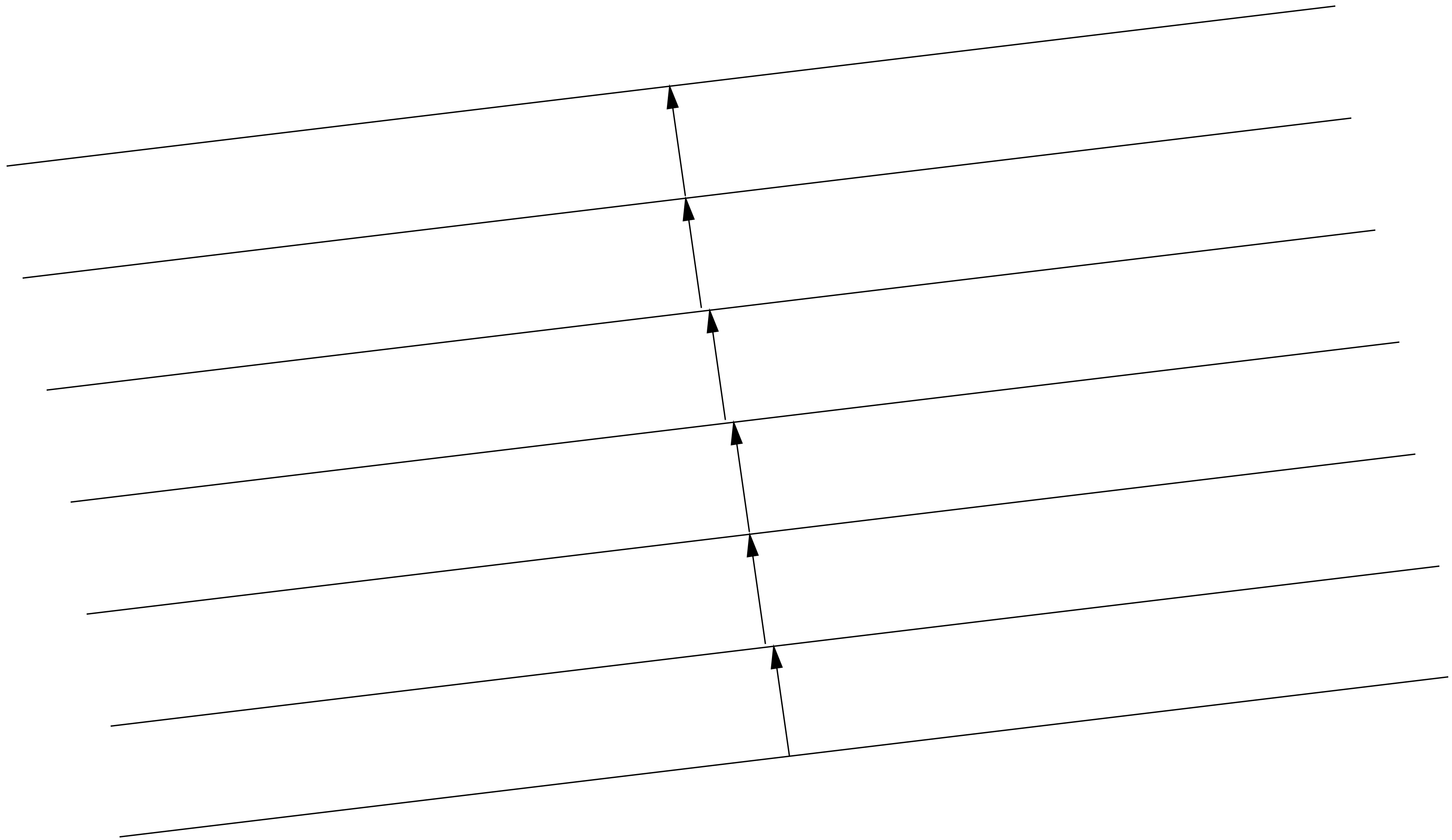
Génération des clés

Choisir u aléatoirement dans $B(0,1)$;

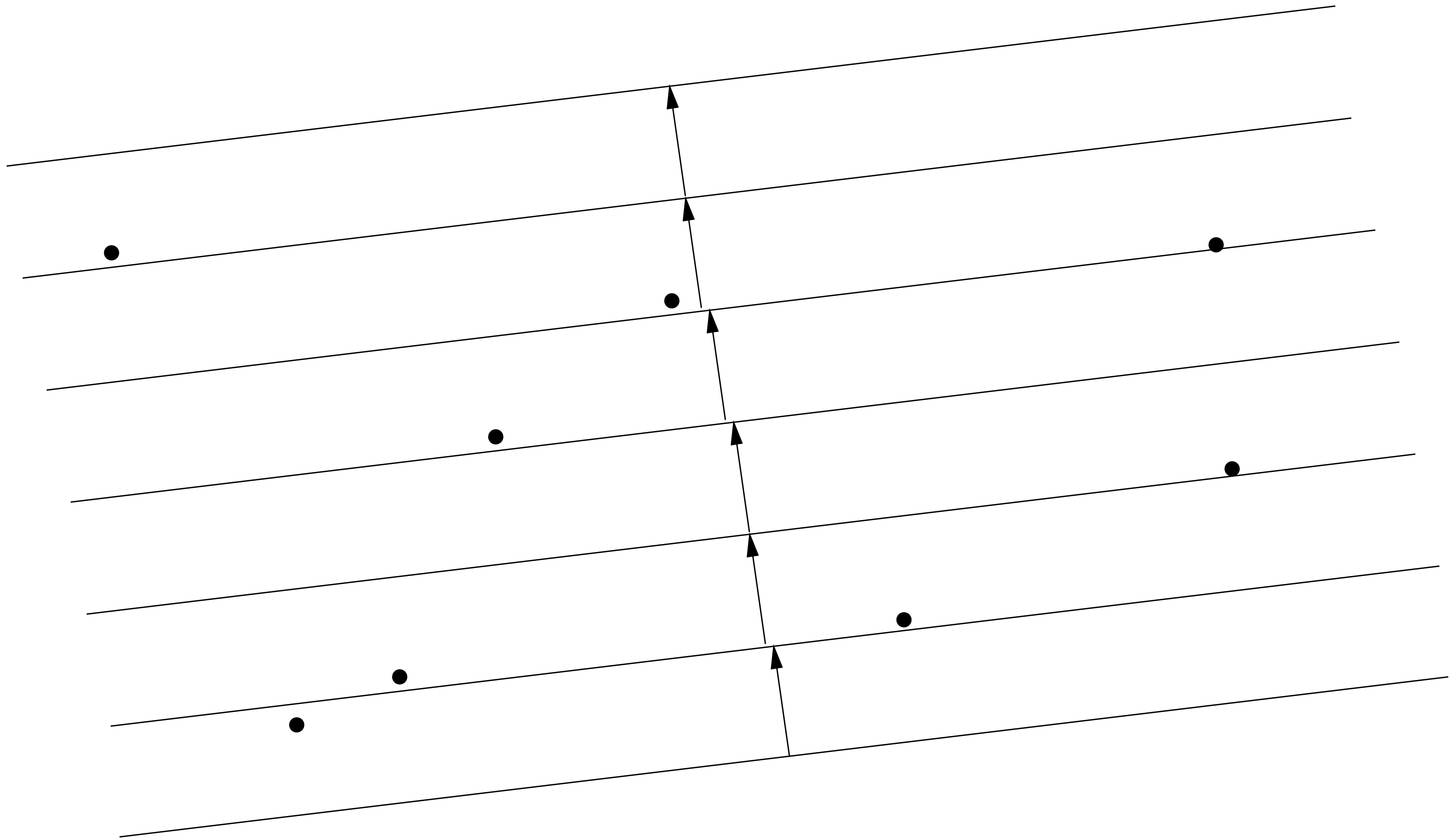
Construire $X = \{x \in \mathbb{R}^n \mid \ < x, u > \in \mathbb{Z}\}$;

Choisir m points $v_1, \cdots, v_m$ “proches des hyperplans” ;

Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”



Un cryptosystème “prouvé sûr”

Génération des clés

Choisir u aléatoirement dans $B(0,1)$;

Construire $X = \{x \in \mathbb{R}^n \mid \ < x, u > \in \mathbb{Z}\}$;

Choisir m points $v_1, \cdots, v_m$ “proches des hyperplans” ;

Trouver i_0 t.q.
Largeur$(\mathcal{P}(v_{i_0+1}, \cdots, v_{i_0+n})) \geq n^{-2}\kappa$;

Un cryptosystème “prouvé sûr”

Génération des clés

Choisir u aléatoirement dans $B(0,1)$;

Construire $X = \{x \in \mathbb{R}^n \mid\ < x, u > \in \mathbb{Z}\}$;

Choisir m points $v_1, \cdots, v_m$ “proches des hyperplans” ;

Trouver i_0 t.q. Largeur$(\mathcal{P}(v_{i_0+1}, \cdots, v_{i_0+n})) \geq n^{-2}\kappa$;

u est la clé privée ;

$v_1, \cdots, v_m$ et i_0 forment la clé publique ;

Un cryptosystème “prouvé sûr”

Chiffrer b

Si $b = 0$

1. Choisir dans $\{0, 1\}$ m valeurs aléatoires $\delta_1, \cdots, \delta_m$;
2. $x := \sum_{i=1}^{m} \delta_i v_i$;
3. $x \;:=\; x$ modulo $(\mathcal{P}(v_{i_0+1}, \cdots, v_{i_0+n}))$;

Si $b = 1$

1. Choisir x aléatoirement dans le “gros cube” ;

Retourner x ;

Un cryptosystème “prouvé sûr”

Déchiffrer b

1. calculer $< u, b >= i + \theta$, $i \in \mathbb{Z}$;
2. Si($|\theta| < \frac{1}{n}$) retourner 0
3. Sinon retourner 1

Erreur au déchiffrement

Un cryptosystème prouvé sûr

Théorème

Si un algo A, étant donné une instance aléatoire du cryptosystème, permet de distinguer les chiffrés des 0 de ceux des 1, avec une proba. $> \frac{1}{2}$

Alors on en déduit un algo B qui trouve le n^c-unique plus court vecteur avec une proba. $1 - 2^{-n}$.

Malheureusement...

Cryptosystème impracticable

Forte expansion

Clé publique ($n = 32$, 20 Mo)

Cassé par Nguyen et Stern en 98

Les pistes

Chiffrement par bloc ?

Trouver d'autres connections avec des problèmes dans les graphes.