Signatures Fondées sur les Réseaux Euclidiens: Attaques, Analyse et Optimisations

Soutenance de thèse de doctorat, spécialité informatique

Léo Ducas-Binda

Université Paris Diderot, École Normale Supérieure

12 Novembre 2013

Signatures Fondées sur les Réseaux Euclidiens : Attaques, Analyse et Optimisations

Introduction

1. Signatures numériques
2. Réseaux euclidiens : géométrie et algorithmes

Contributions

3. Signatures heuristiques et attaques
4. Analyse et optimisation de l'ÉGD
5. BLISS : Un schéma de signature efficace

Conclusion

6. Conclusion et perspectives

Plan

Introduction

1. Signatures numériques
2. Réseaux euclidiens : géométrie et algorithmes

Contributions

3. Signatures heuristiques et attaques
4. Analyse et optimisation de l'ÉGD
5. BLISS : Un schéma de signature efficace

Conclusion

6. Conclusion et perspectives

Signature numérique

Analogue **numérique** de la signature manuscrite ou du **paraphe**.
Elle doit être :

- **inimitable**
- **vérifiable** par tous.

Signature numérique

Analogue **numérique** de la signature manuscrite ou du **paraphe**.
Elle doit être :

- **inimitable**
- **vérifiable** par tous.

Signature numérique

Analogue **numérique** de la signature manuscrite ou du **paraphe**.
Elle doit être :

- **inimitable**
- **vérifiable** par tous.

à Paris, le 12 Nov. 2013

Empêcher le copier-coller

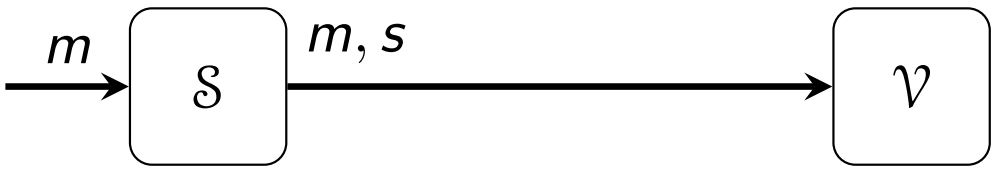
Dans l'univers numérique, tout est **copiable**.

- la signature numérique ne peut être simplement apposée

Empêcher le copier-coller

Dans l'univers numérique, tout est **copiable**.

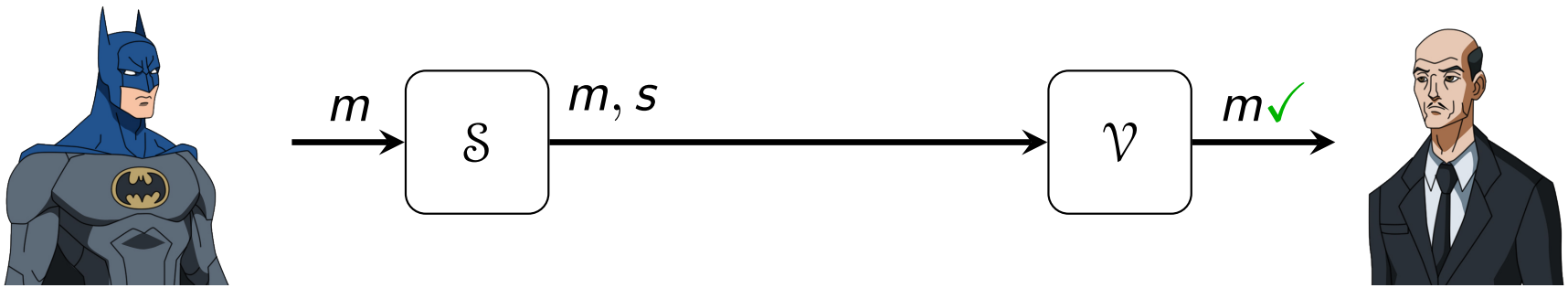
- la signature numérique ne peut être simplement apposée
- elle doit être liée mathématiquement au contenu du message.



Empêcher le copier-coller

Dans l'univers numérique, tout est **copiable**.

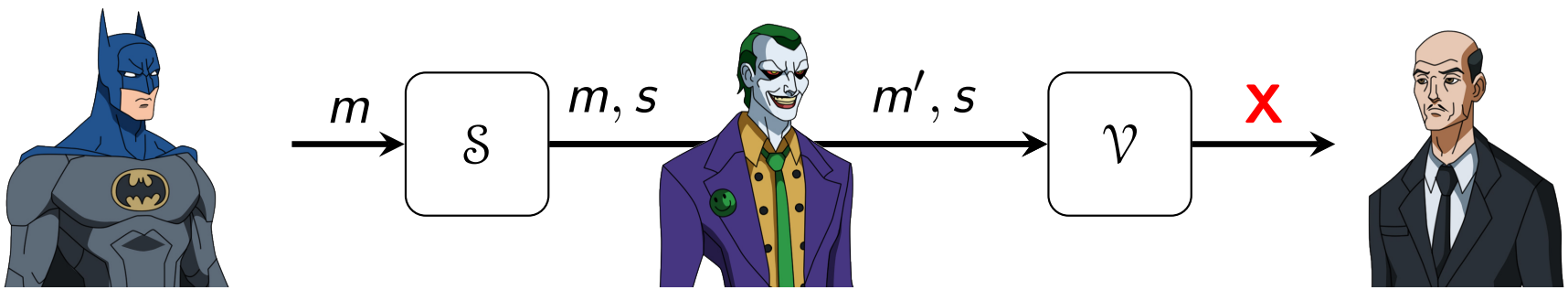
- la signature numérique ne peut être simplement apposée
- elle doit être liée mathématiquement au contenu du message.



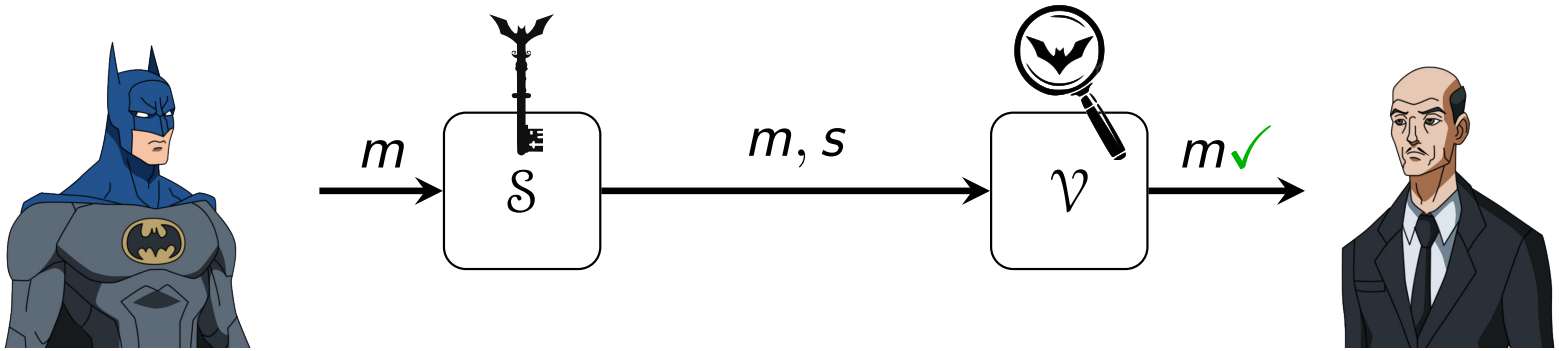
Empêcher le copier-coller

Dans l'univers numérique, tout est **copiable**.

- la signature numérique ne peut être simplement apposée
- elle doit être liée mathématiquement au contenu du message.



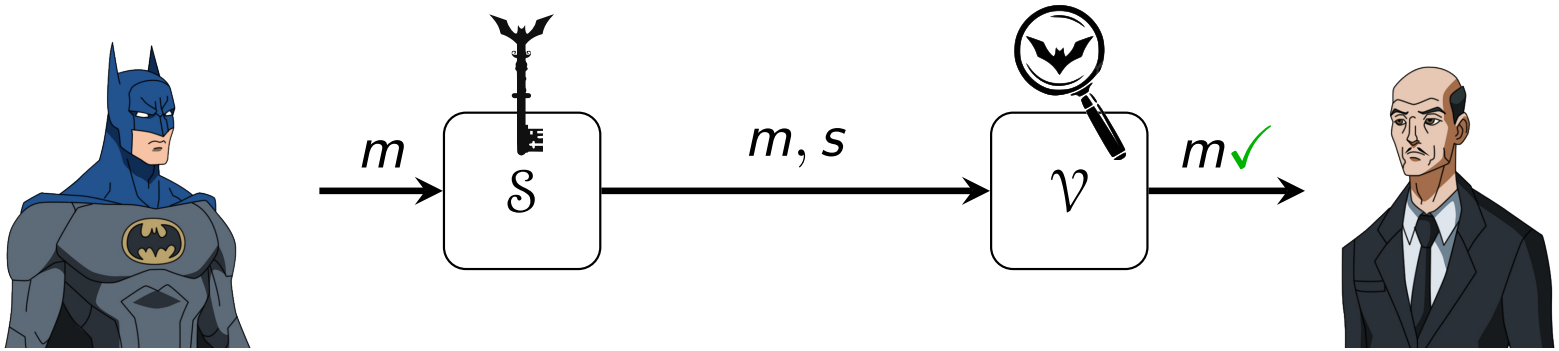
Clef privée / clef publique



La Signature $\mathcal{S}$ requiert une **clef privée** .

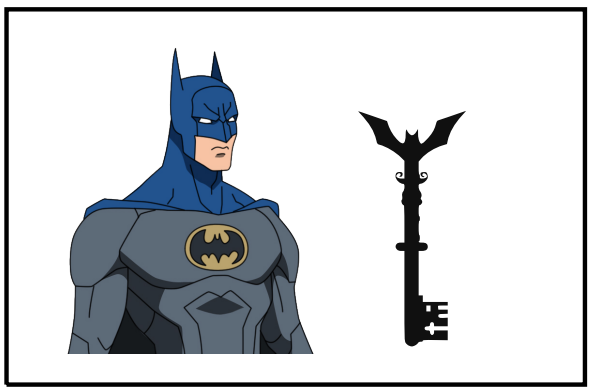
La vérification $\mathcal{V}$ requiert une **clef publique** .

Clef privée / clef publique

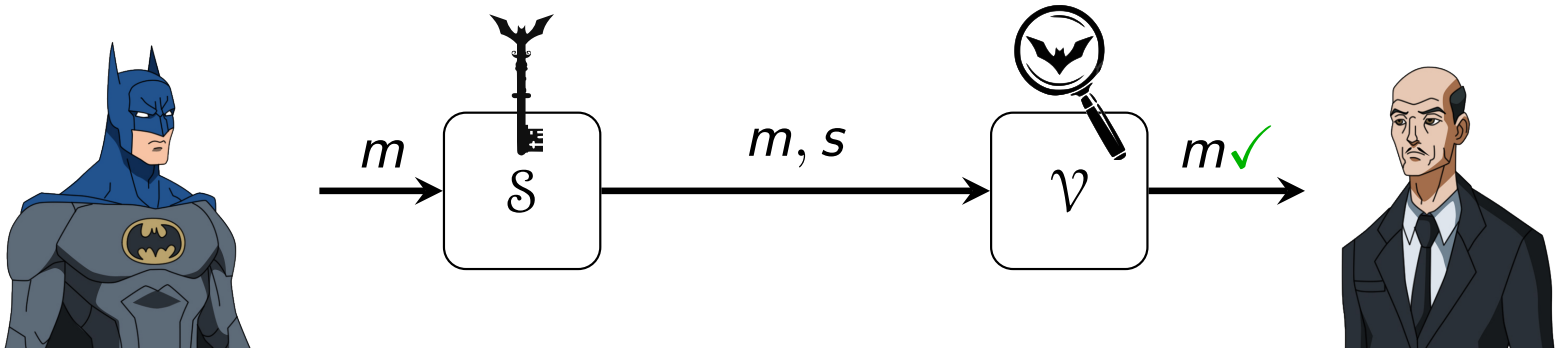


La Signature $\mathcal{S}$ requiert une **clef privée** .

La vérification $\mathcal{V}$ requiert une **clef publique** .

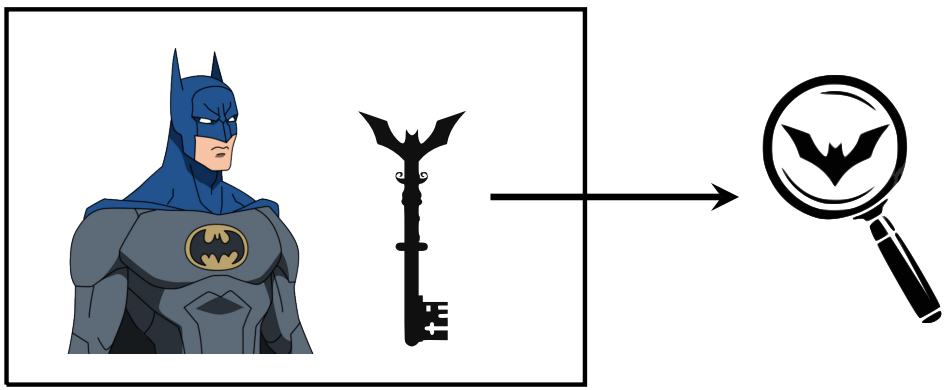


Clef privée / clef publique

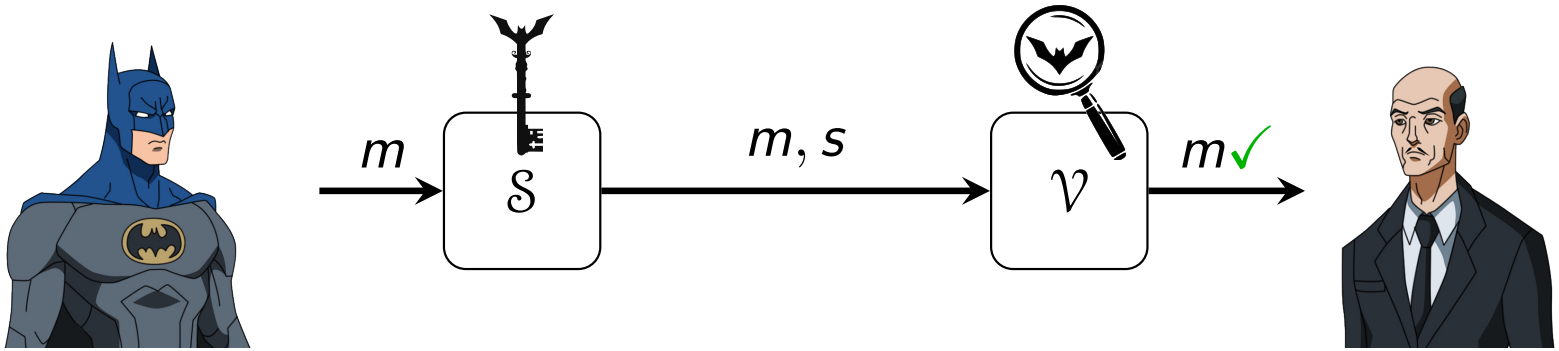


La Signature $\mathcal{S}$ requiert une **clef privée** .

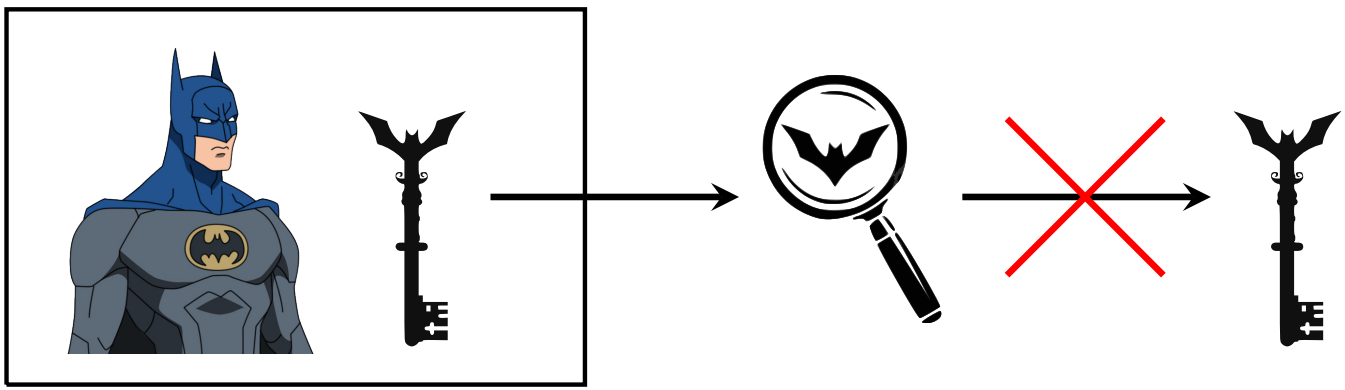
La vérification $\mathcal{V}$ requiert une **clef publique** .



Clef privée / clef publique



La Signature $\mathcal{S}$ requiert une **clef privée** .
La vérification $\mathcal{V}$ requiert une **clef publique** .



En particulier, la transformation **clef privée** $\rightarrow$ **clef publique** doit être à **sens unique**.

Applications

Les signatures **authentifient** la source d'une information.
Applications :

- mise à jour de logiciel
  protection contre l'injection de malware
- protection du routage global sur internet
  protocoles **DNSec**, **BGPsec**
- infrastructure à clef publique pour le chiffrement
  protocole **https**
- ...

Applications

Les signatures **authentifient** la source d'une information.
Applications :

- mise à jour de logiciel
  protection contre l'injection de malware
- protection du routage global sur internet
  protocoles **DNSec**, **BGPsec**
- infrastructure à clef publique pour le chiffrement
  protocole **https**
- ...

- administration entièrement informatisée ?

Plan

Introduction

1 Signatures numériques

2 Réseaux euclidiens : géométrie et algorithmes

Contributions

3 Signatures heuristiques et attaques

4 Analyse et optimisation de l'ÉGD

5 BLISS : Un schéma de signature efficace

Conclusion

6 Conclusion et perspectives

Avantages de la cryptographie fondée sur les réseaux

Meilleures garanties de sécurité :

- réductions pire-cas/cas-moyen
- résistances aux algorithmes quantiques
- problèmes étudiés au delà de la cryptographie.

Plus souple :

- Chiffrement homomorphe, fonctionnel, encodage multilinéaire.

Plus efficace :

- complexité en $\tilde{O}(n^2)$ voir $\tilde{O}(n)$
- parallélisable
- opérations modulo un petit entier q.

Avantages de la cryptographie fondée sur les réseaux

Meilleures garanties de sécurité :

- réductions pire-cas/cas-moyen
- résistances aux algorithmes quantiques
- problèmes étudiés au delà de la cryptographie.

Plus souple :

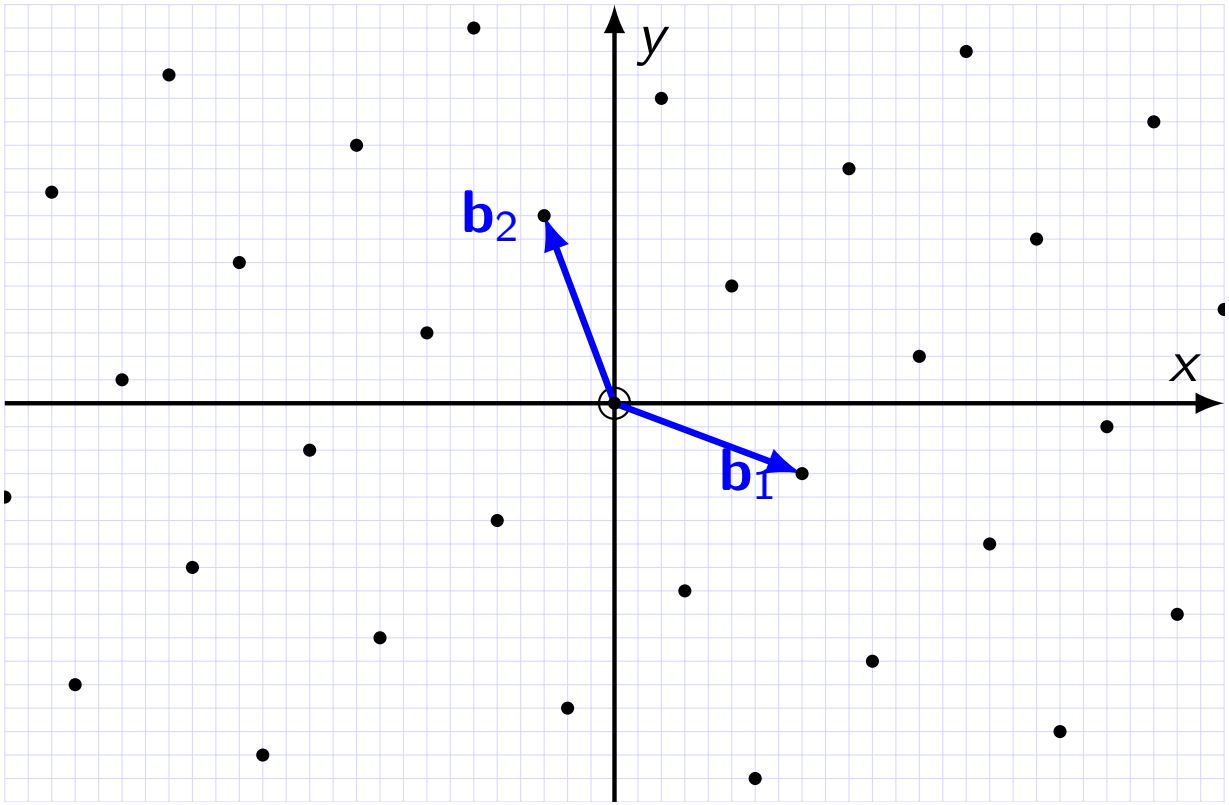
- Chiffrement homomorphe, fonctionnel, encodage multilinéaire.

Plus efficace :

- complexité en $\tilde{O}(n^2)$ voir $\tilde{O}(n)$
- parallélisable
- opérations modulo un petit entier q.

En théorie . . .

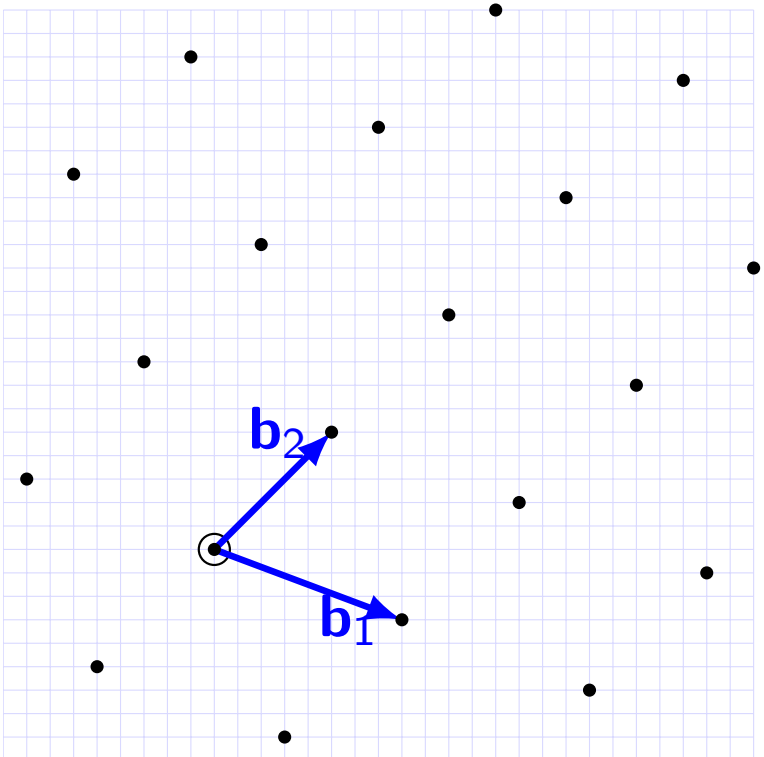
Réseaux euclidiens



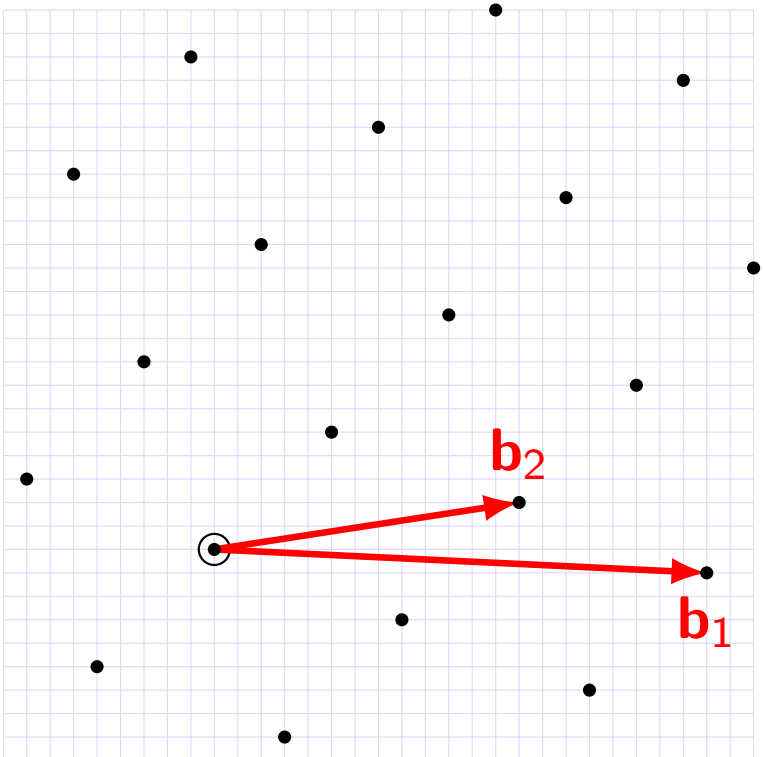
Définition (Réseau euclidien)

Un réseau euclidien L est un sous-groupe discret d'un espace vectoriel de dimension fini $\mathbb{R}^n$.

Bases d'un réseau



Bonne Base **B** de L



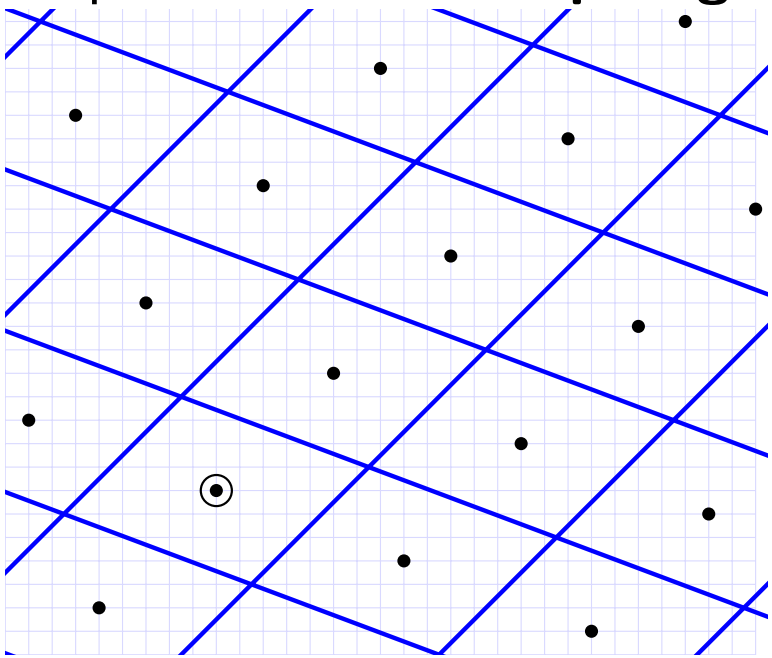
Mauvaise Base **M** de L

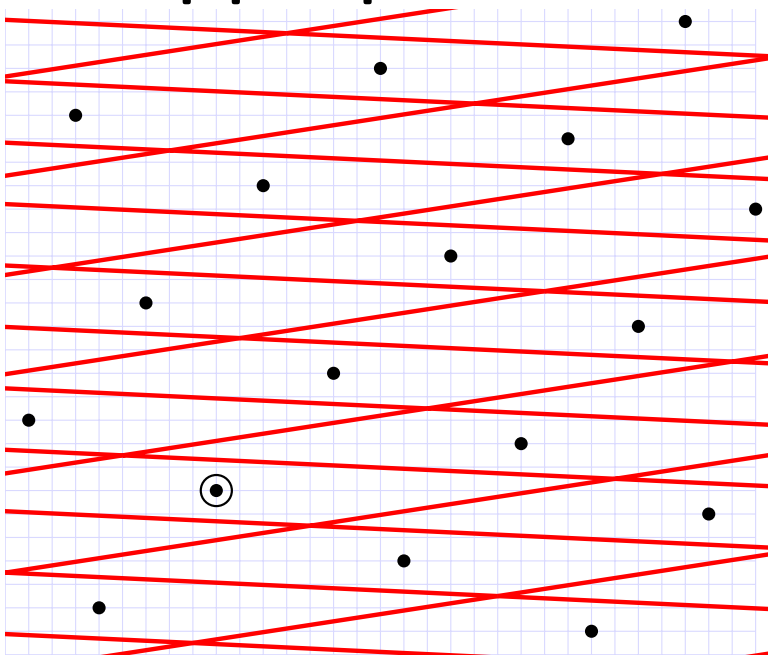
B $\rightarrow$ **M** : facile (randomisation) ;

M $\rightarrow$ **B** : difficile (réduction de réseau).

Bases et domaines fondamentaux

Chaque base définit un **pavage parallélépipédique**.

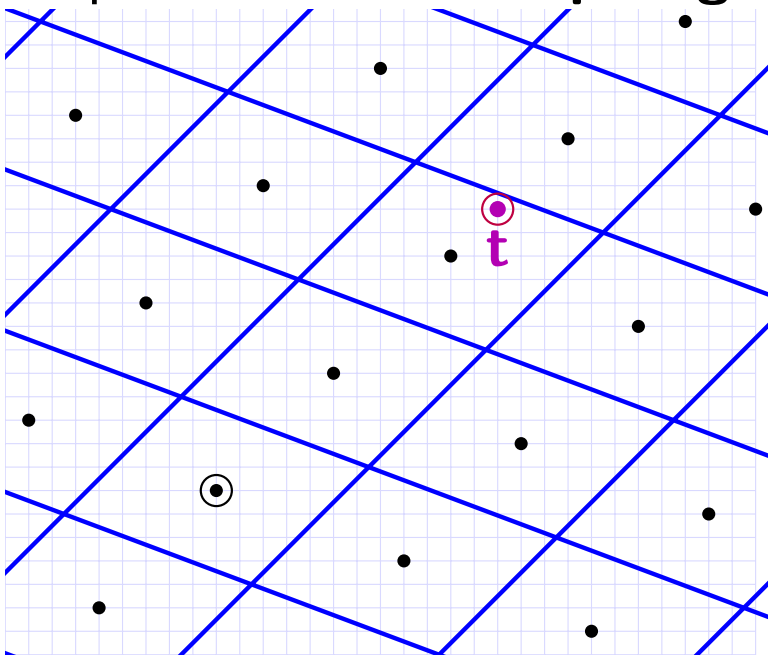


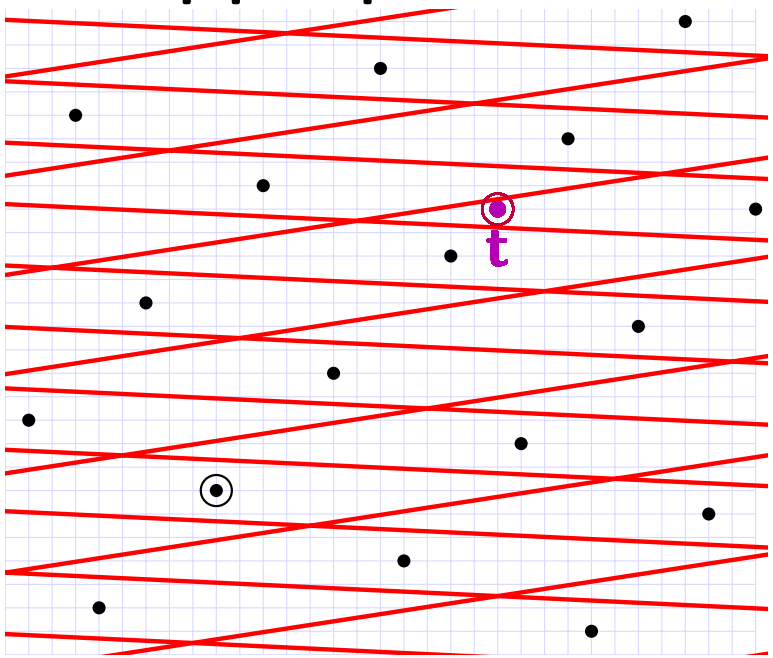


Algorithmes de Babai :

Bases et domaines fondamentaux

Chaque base définit un **pavage parallélépipédique**.



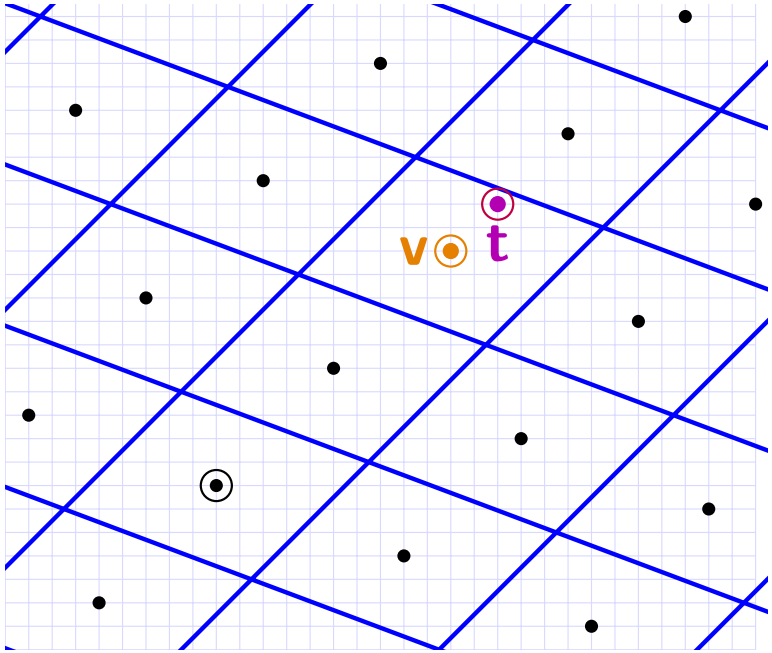


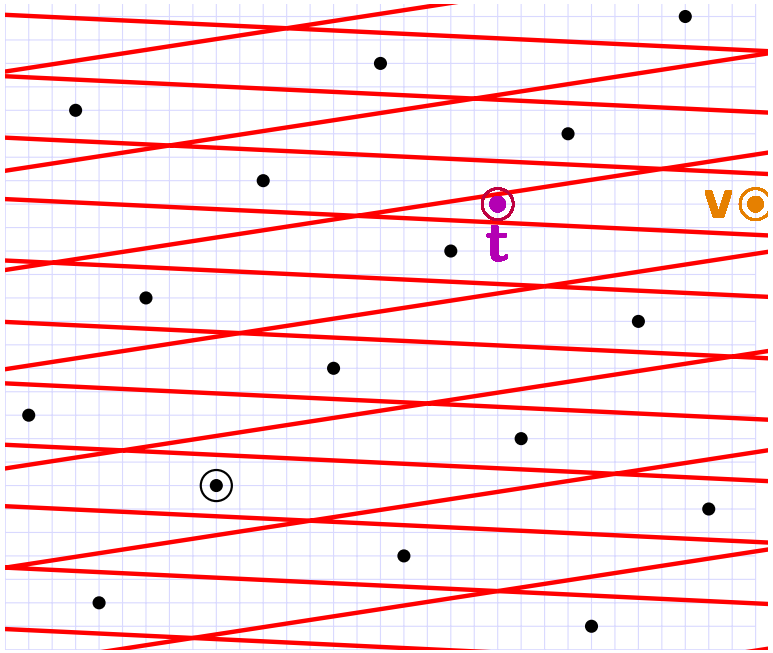
Algorithmes de Babai :

- étant donné un point **t**

Bases et domaines fondamentaux

Chaque base définit un **pavage parallélépipédique**.

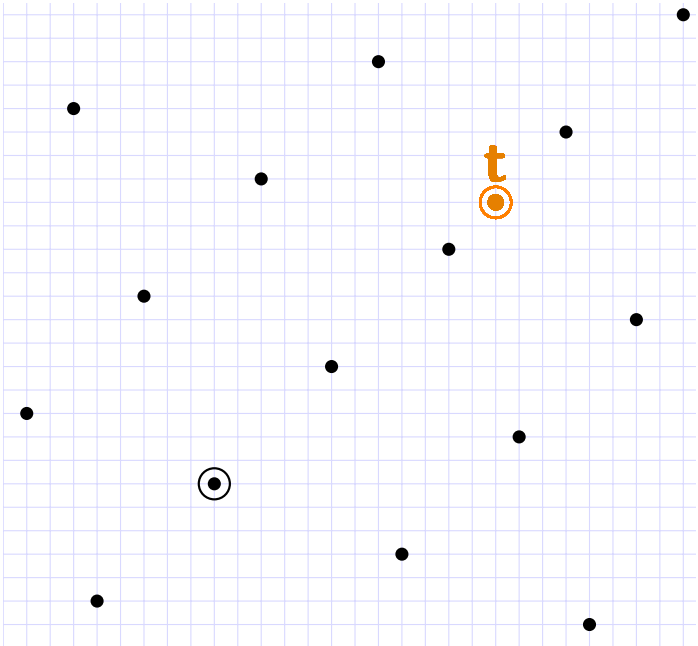




Algorithmes de Babai :

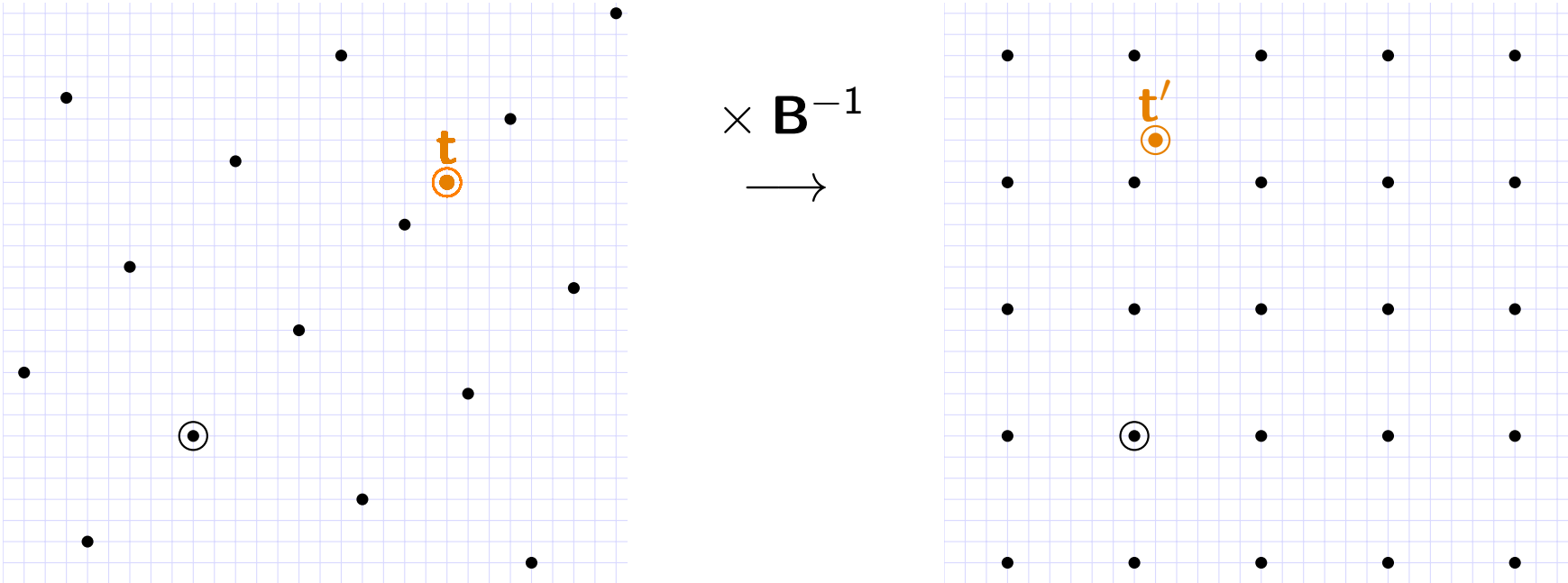
- étant donné un point **t**
- trouve le point **v** $\in L$ dans le même parallélépipède.

Arrondi de Babai



Algorithme de Babai, arrondi simple :

Arrondi de Babai

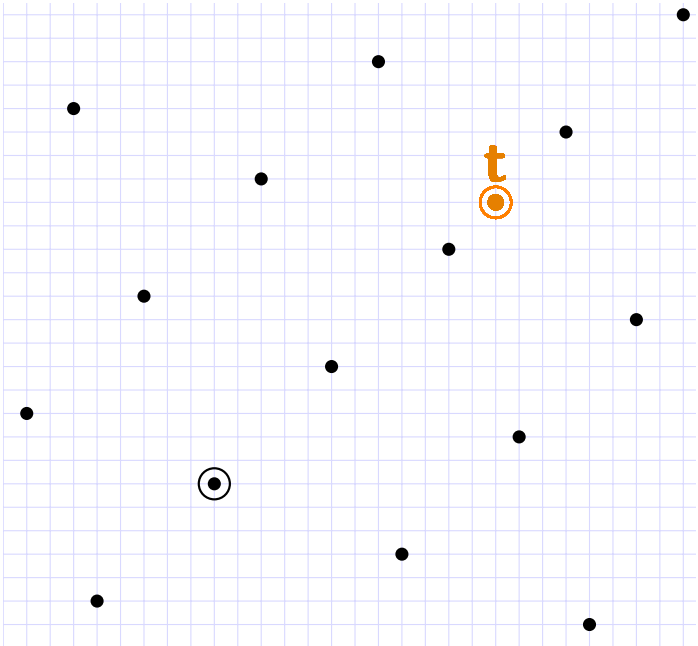


Algorithme de Babai, arrondi simple :

- utiliser la base $\mathbf{B}$ pour se ramener au réseau $\mathbb{Z}^n$ $(\times \mathbf{B}^{-1})$

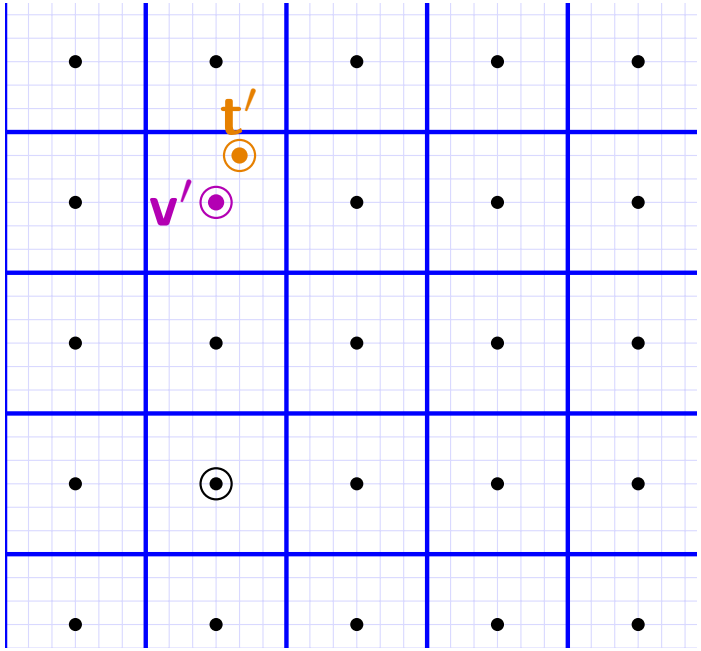
$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t};$$

Arrondi de Babai



$\times \mathbf{B}^{-1}$

$\longrightarrow$

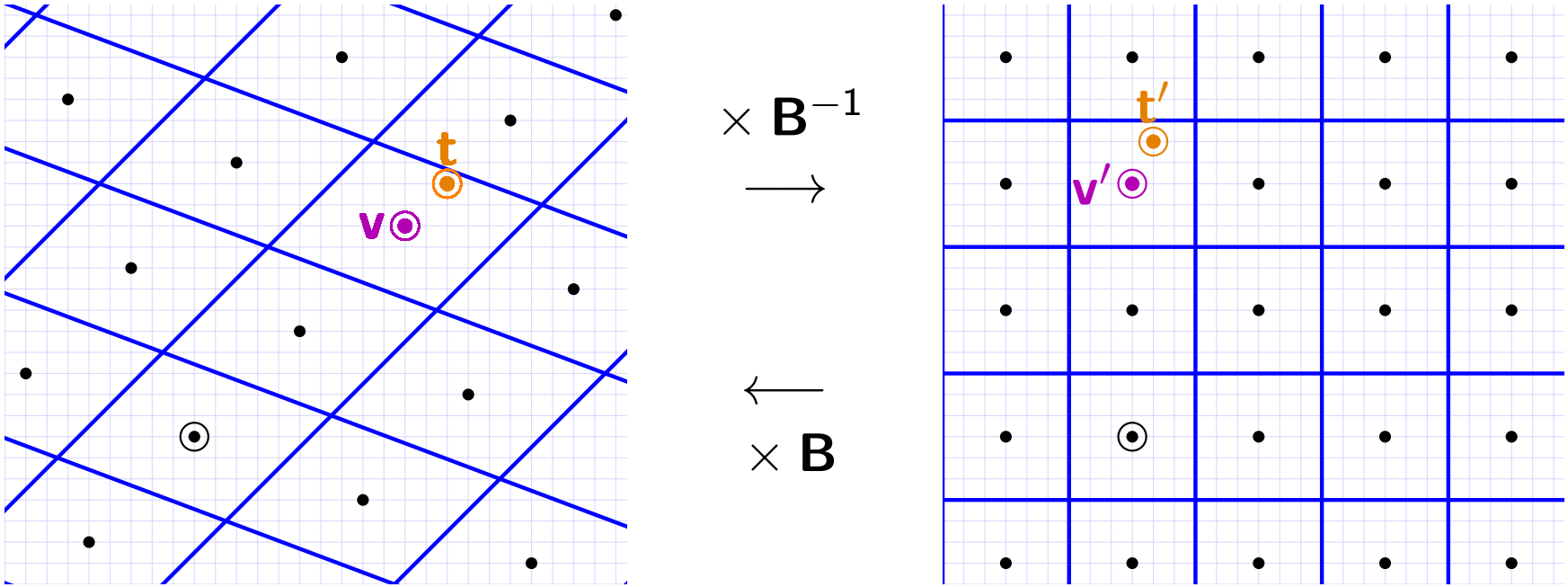


Algorithme de Babai, arrondi simple :

- utiliser la base $\mathbf{B}$ pour se ramener au réseau $\mathbb{Z}^n$ $(\times \mathbf{B}^{-1})$
- arrondir chaque coordonnée (découpage carré)

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil;$$

Arrondi de Babai



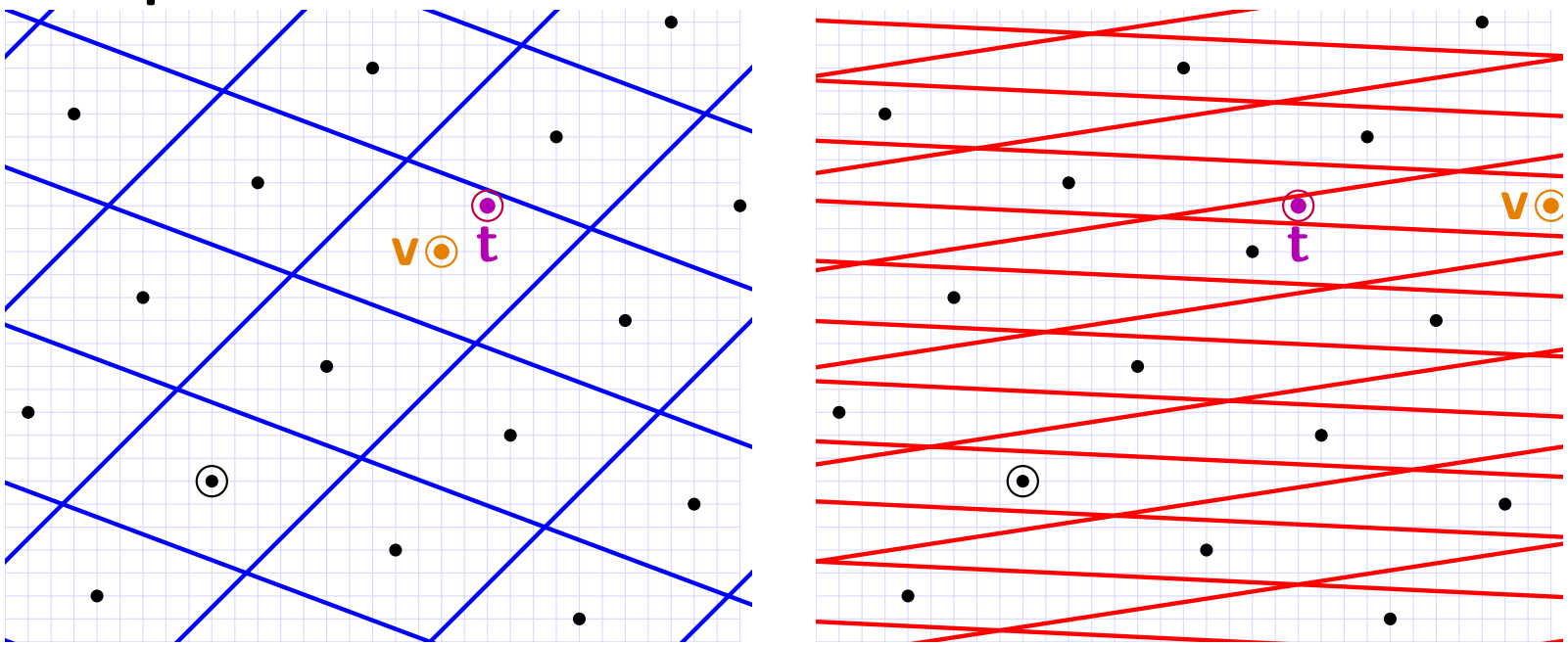
Algorithme de Babai, arrondi simple :

- utiliser la base $\mathbf{B}$ pour se ramener au réseau $\mathbb{Z}^n$ $(\times \mathbf{B}^{-1})$
- arrondir chaque coordonnée (découpage carré)
- revenir au réseau L $(\times \mathbf{B})$

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Trouver des vecteurs proches

Muni d'une bonne base **B** l'algorithme de Babai permet de trouver des vecteurs proches.
Muni seulement d'une mauvaise base **M**, trouver un vecteur proche est un **problème difficile**.



Cette différence permet de fonder la **cryptographie asymétrique**.

Plan

Introduction

1 Signatures numériques

2 Réseaux euclidiens : géométrie et algorithmes

Contributions

3 Signatures heuristiques et attaques

4 Analyse et optimisation de l'ÉGD

5 BLISS : Un schéma de signature efficace

Conclusion

6 Conclusion et perspectives

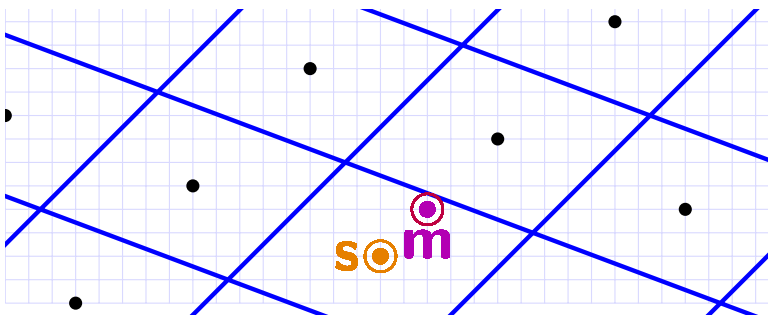
Schéma de signature de [GGH97]

Signature

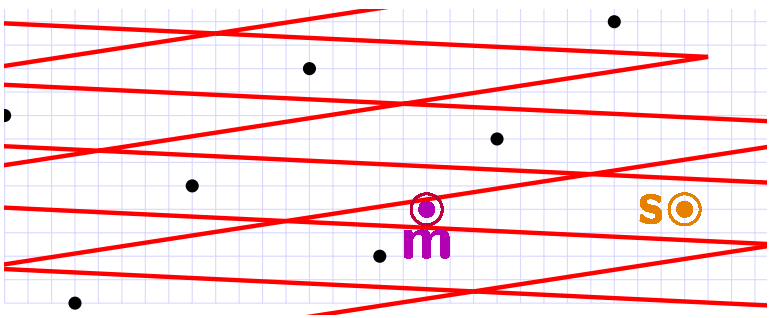
- hacher le message comme un vecteur aléatoire $\mathbf{m}$.
- appliquer l'algorithme Babai avec $\mathbf{B}$:
  trouver $\mathbf{s} \in L$ proche de $\mathbf{m}$.

Vérification

- vérifier que $\mathbf{s} \in L$ en utilisant $\mathbf{M}$
- et que $\mathbf{m}$ est proche de $\mathbf{s}$.



Signature correcte (proche)



Signature incorrecte (éloignée)

GGH, NTRUSIGN, attaque et contre-mesures

NTRUSIGN est une version optimisée de la signature GGH : réseaux structurés et q-aire, base secrète très courte...

Faiblesse : chaque signature GGH laisse fuir un peu d'information sur la clef secrète

$\Rightarrow$ attaque pratique avec 400 signatures [NR06].

GGH, NTRUSIGN, attaque et contre-mesures

NTRUSIGN est une version optimisée de la signature GGH : réseaux structurés et q-aire, base secrète très courte...

Faiblesse : chaque signature GGH laisse fuir un peu d'information sur la clef secrète

$\Rightarrow$ attaque pratique avec 400 signatures [NR06].

Contre-mesures proposées :

- ? la technique originale de perturbation [HNHGSW03]
- ? la technique de déformation [HWH08]
- ✓ l'échantillonnage gaussien discret [Kle00, GPV08].

GGH, NTRUSIGN, attaque et contre-mesures

NTRUSIGN est une version optimisée de la signature GGH : réseaux structurés et q-aire, base secrète très courte...

Faiblesse : chaque signature GGH laisse fuir un peu d'information sur la clef secrète

$\Rightarrow$ attaque pratique avec 400 signatures [NR06].

Contre-mesures proposées :

- ? la technique originale de perturbation [HNHGSW03]
- ? la technique de déformation [HWH08]
- ✓ l'échantillonnage gaussien discret [Kle00, GPV08].

Des faiblesses théoriques [HNHGSW03, MPSW] suggérées.

Question : une attaque en pratique ?

GGH, NTRUSIGN, attaque et contre-mesures

NTRUSIGN est une version optimisée de la signature GGH : réseaux structurés et q-aire, base secrète très courte...

Faiblesse : chaque signature GGH laisse fuir un peu d'information sur la clef secrète

$\Rightarrow$ attaque pratique avec 400 signatures [NR06].

Contre-mesures proposées :

?X la technique originale de perturbation [HNHGSW03]

?X la technique de déformation [HWH08]

✓ l'échantillonnage gaussien discret [Kle00, GPV08].

Des faiblesses théoriques [HNHGSW03, MPSW] suggérées.

Question : une attaque en pratique ?

Modèles de fuites et attaques statistiques

Support de la distribution de $\mathbf{m} - \mathbf{s}$ pour GGH et NTRUSIGN :

Brut (sans perturbation)	Avec perturbation
un **parallélépipède** en dim. n engendré par n vecteurs indep. $\mathcal{P} = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in [-1, 1]\}$	un **zonotope** en dim. n engendré par $m \geqslant n$ vecteurs $\mathcal{Z} = \{\sum_{i=1}^{m} x_i \mathbf{b}_i : x_i \in [-1, 1]\}$
Paral. de Babai	Somme de Paral. de Babai

Le problème à résoudre

Pour forger NTRUSIGN **brut**, il a fallu résoudre le problème :

Apprentissage d'un parallélépipède $\mathcal{P}(\mathbf{V})$ [NR06].

- **pour** une matrice $n \times n$ inversible $\mathbf{V}$
- **étant donnés** des échantillons $\mathbf{x} = \mathbf{V} \cdot \mathbf{y}$ où $\mathbf{y} \leftarrow [-1, 1]^n$
- **retrouver** une (ou toute) colonne de la matrice $\mathbf{V}$.

Avec perturbation, nous résolvons le problème généralisé :

Apprentissage d'un zonotope $\mathcal{Z}(\mathbf{V})$ [D. Nguyen '12]

- **pour** une matrice $n \times m$ $\mathbf{V}$, $m \geqslant n$
- **étant donnés** des échantillons $\mathbf{x} = \mathbf{V} \cdot \mathbf{y}$ où $\mathbf{y} \leftarrow [-1, 1]^m$
- **retrouver** une (ou toute) colonne de la matrice $\mathbf{V}$.

L'attaque de [NR06], 1ère étape

L'attaque s'appuie sur des mesures statistiques, les moments d'ordre 2 et 4 sur la sphère unité.

$$\forall v \in \mathbb{S}_n, \quad \text{mom}_{k,\mathcal{D}}(\mathbf{v}) = \mathbb{E}_{\mathbf{x} \leftarrow \mathcal{D}} \left[\langle \mathbf{x}, \mathbf{v} \rangle^k \right]$$

Isotropisation : L'attaque originale se ramène d'abord au cas de l'hypercube, en mesurant $\text{mom}_{2,\mathcal{P}}$ (*i.e.* la matrice de covariance).

.

$\rightarrow$

L'attaque de [NR06], $2^{\text{ème}}$ étape

Théorème ([NR06], Minima de $\text{mom}_{4,\mathcal{C}}$)

Si $\mathcal{C} = \mathcal{P}(\mathbf{B})$ est un hypercube, alors les minima locaux de $\text{mom}_{4,\mathcal{C}}$ sont exactement les vecteurs $\pm\mathbf{b}_1 \cdots \pm \mathbf{b}_n$.

$\Rightarrow$ $2^{\text{ème}}$ étape : Trouver ces minima à l'aide d'une **descente de gradient**.

Apprendre un zonotope isotrope ?

Observation

Les minima de $\mathrm{mom}_{4,\mathcal{Z}(\mathbf{B})}$, ne sont plus exactement les vecteurs $\pm\mathbf{b}_1 \cdots \pm \mathbf{b}_m$.

Cependant, certains minima en sont **proches** !

En particulier pour les $\mathbf{b}_i$ **presque orthogonaux** aux autres $\mathbf{b}_j$, $j \neq i$.

Généralisation du théorème de [NR06]

Nous généralisons le théorème précedent :

Théorème (Minima de $\mathrm{mom}_{4,\mathcal{Z}}$ [D. Nguyen '12])

Si $\mathcal{Z} = \mathcal{Z}(\mathbf{B})$ est un zonotope isotrope, et si les vecteurs $\pm\mathbf{b}_1 \cdots \pm \mathbf{b}_m$ sont **quasi-orthogonaux**, *alors chaque $\pm\mathbf{b}_i$ est proche d'un minimum local de $\mathrm{mom}_{4,\mathcal{Z}}$.*

Méthode de preuve : Décomposition de la fonction $\mathrm{mom}_{4,\mathcal{Z}}$:

$$\mathrm{mom}_{4,\mathcal{Z}}(\mathbf{v}) = \tfrac{1}{3} - \tfrac{2}{15}\sum_i f_{\mathbf{b}_i}(\mathbf{v}) \quad \text{où} \quad f_{\mathbf{b}}(\mathbf{v}) = \langle \mathbf{b}, \mathbf{v} \rangle^4$$

- **quasi-orthogonaux** : bornes sur $\max_{i \neq j} |\langle \mathbf{b}_i, \mathbf{b}_j \rangle|$
- appliquer ces bornes aux dérivées d'ordre 1 et 2 de $f_{\mathbf{b}_i}$ en $\mathbf{b}_j$
- conclure par le théorème de **Taylor-Lagrange vectoriel**.

Nos résultats en pratique

Ajout d'une 3ème étape pour corriger les erreurs
algorithme **BDD** spécialisé pour NTRUSIGN.

Attaque réussie en pratique contre NTRUSIGN avec paramètres standards : récupération totale de la clef secrète

- avec ≈ 4000 signatures malgré la contre-mesure originale de perturbation ;
- avec ≈ 2000 signatures malgré la contre-mesure de déformation.

Conclusion : Il faut utiliser des techniques à la **sécurité prouvée**.
$\Rightarrow$ L'échantillonnage gaussien discret.

Plan

Introduction

1 Signatures numériques

2 Réseaux euclidiens : géométrie et algorithmes

Contributions

3 Signatures heuristiques et attaques

4 Analyse et optimisation de l'ÉGD

5 BLISS : Un schéma de signature efficace

Conclusion

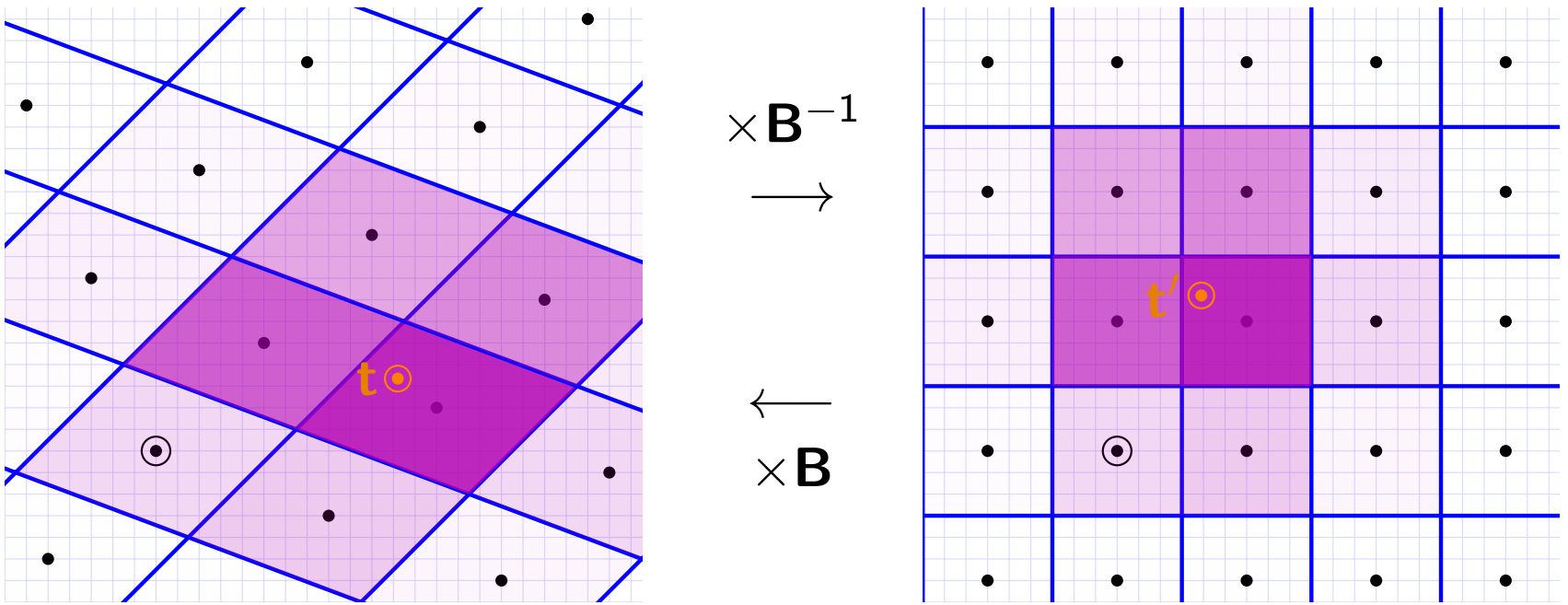
6 Conclusion et perspectives

Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

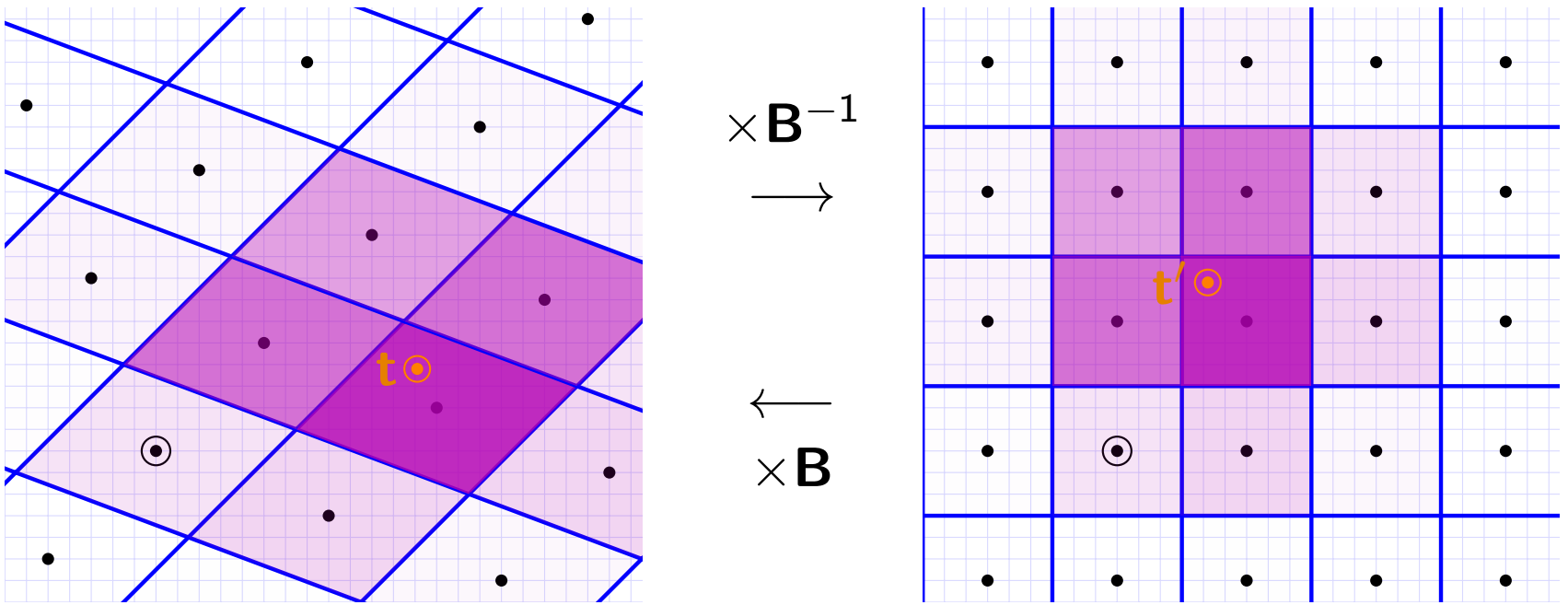


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

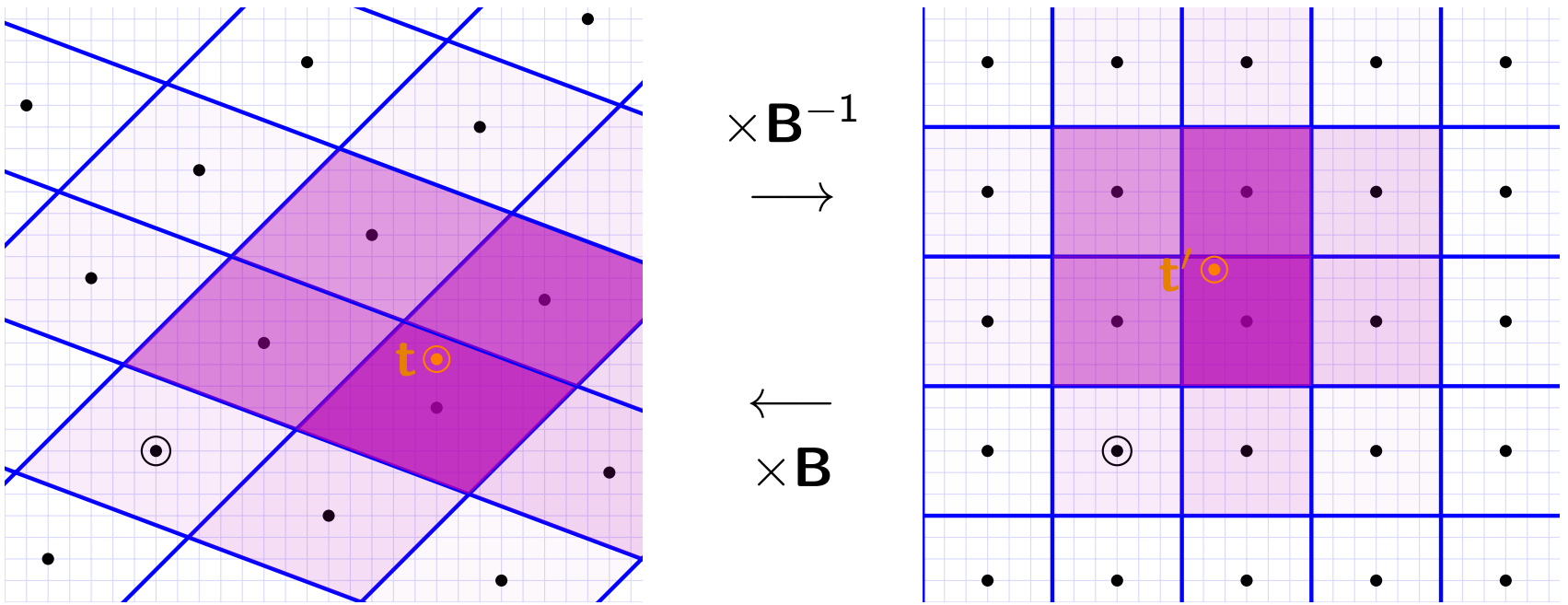


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

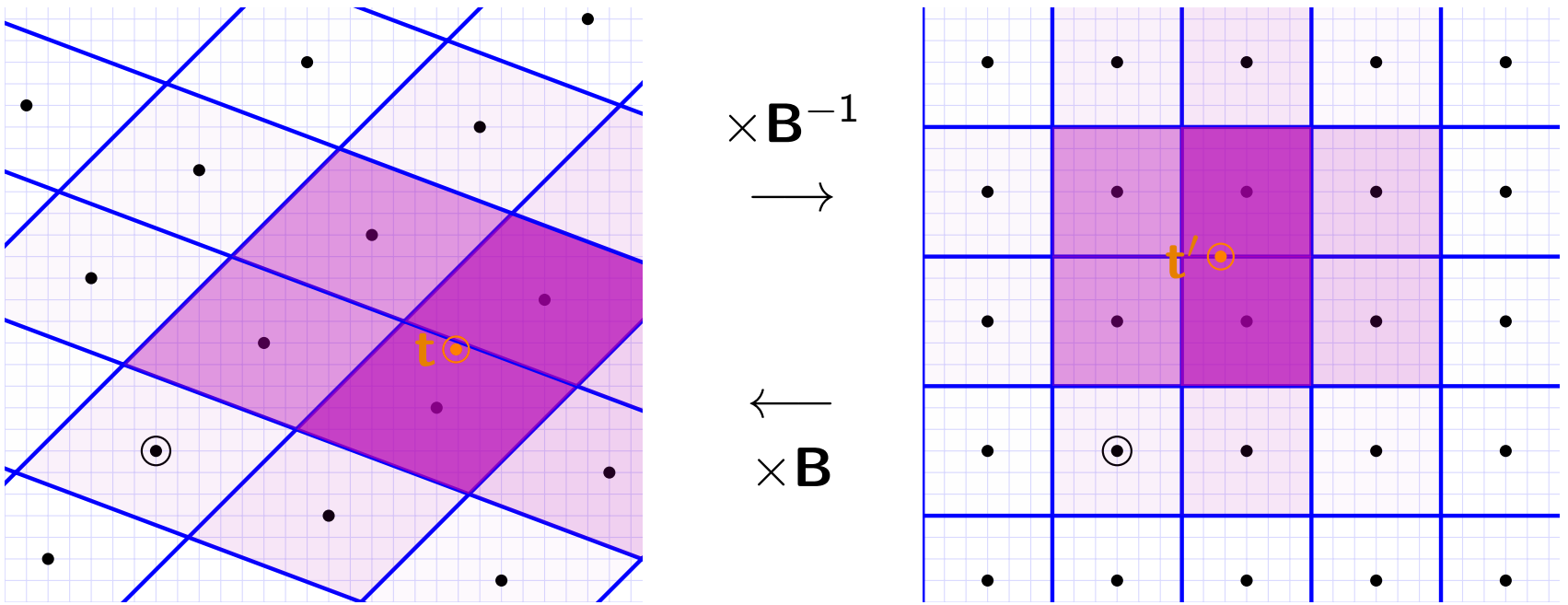


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

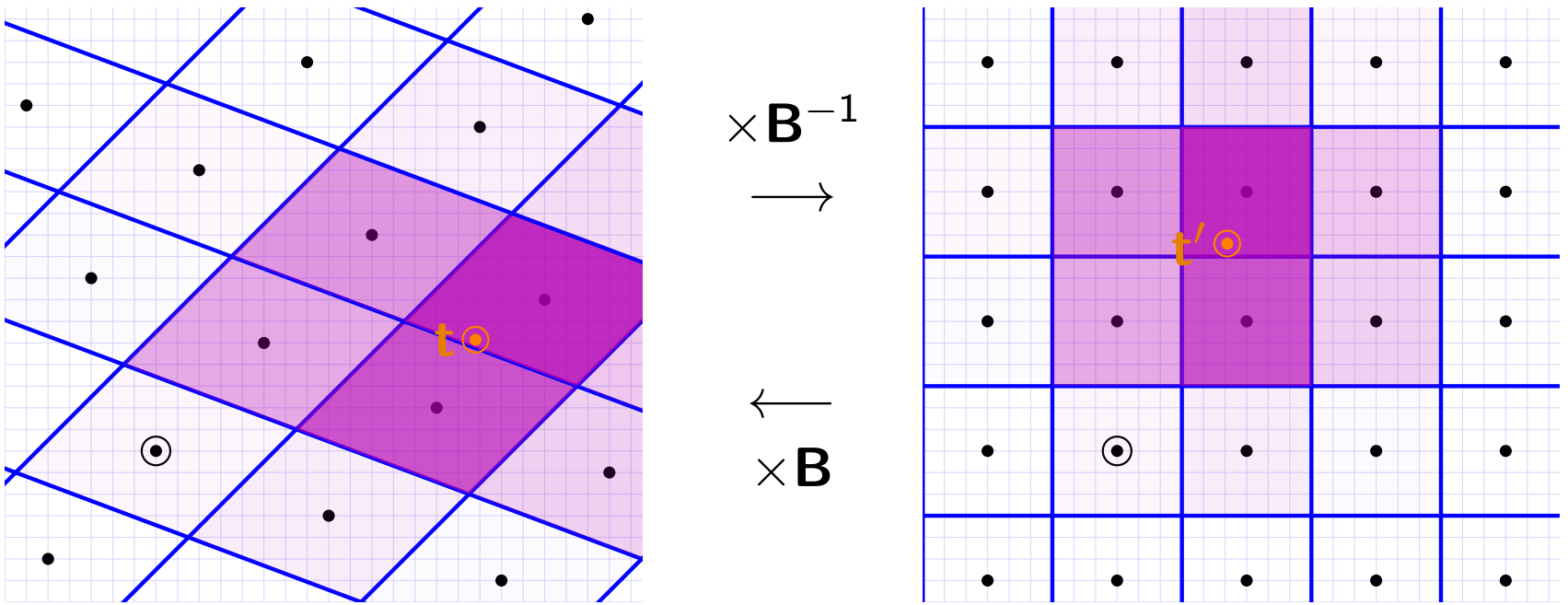


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

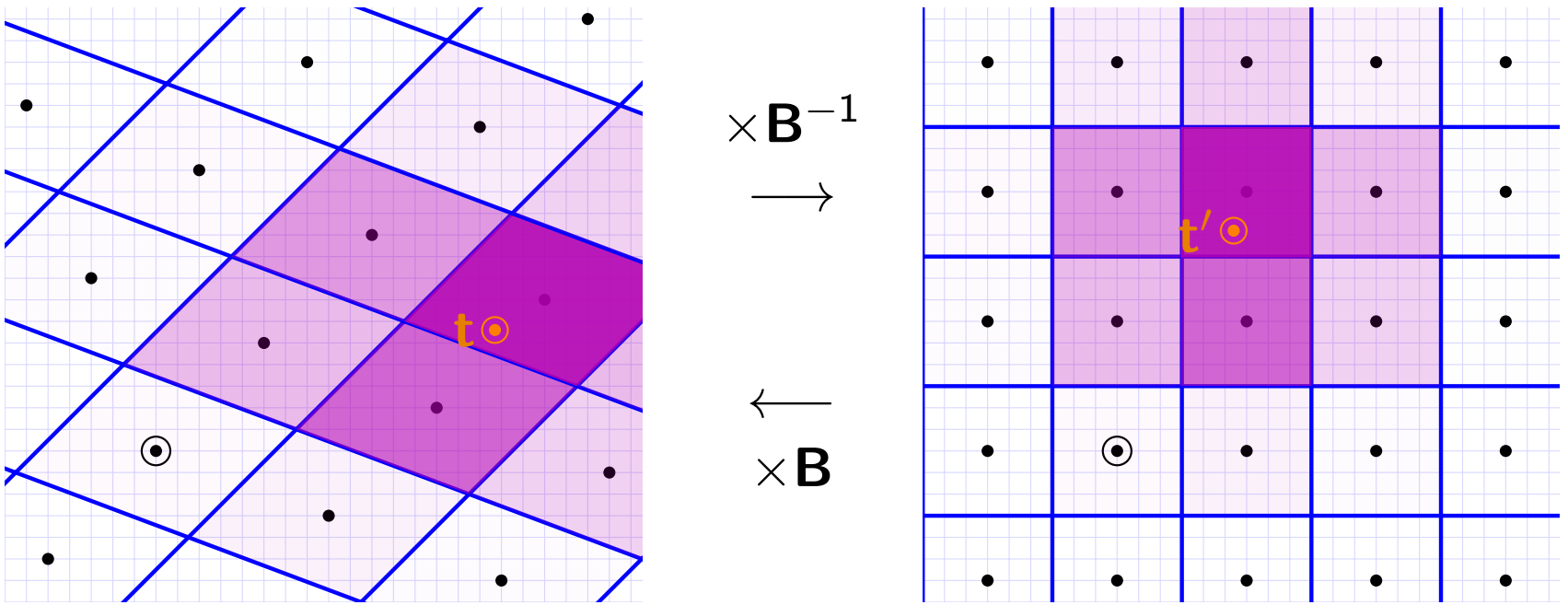


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

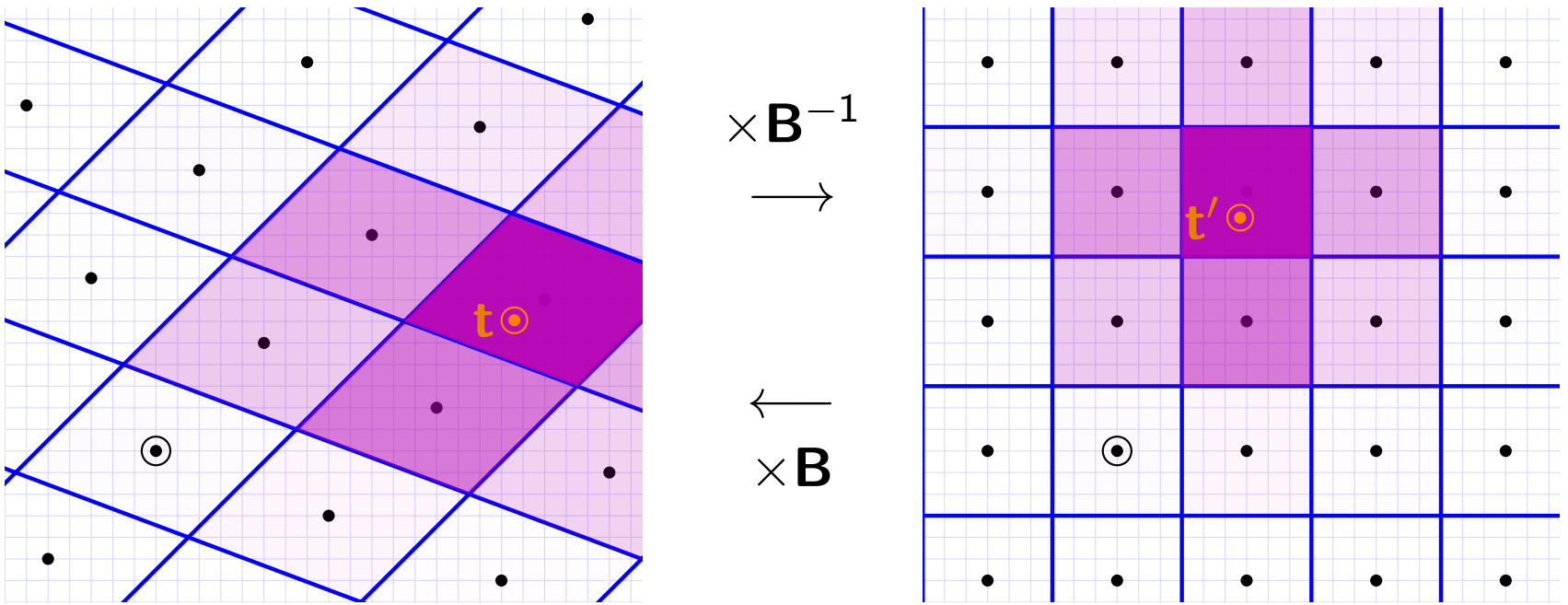


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :

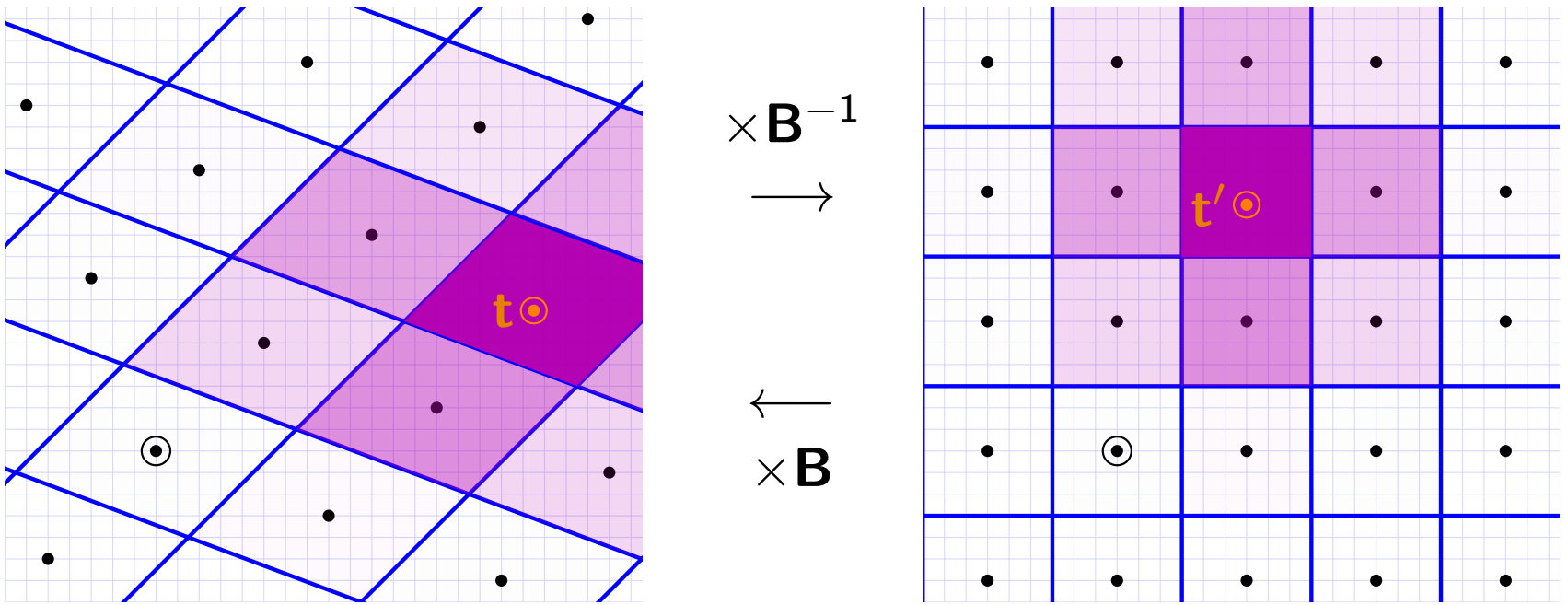


Algorithme de Babai randomisé

Algorithme de Babai :

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \quad \mathbf{v}' = \lfloor \mathbf{t}' \rceil; \quad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$

Idée : Cacher le parallélépipède en randomisant l'arrondi :



Gaussienne discrète et lissage

L'arrondi est remplacé par une gaussienne discrète :

$$\mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'} : \quad \mathbf{v}' \in \mathbb{Z}^n, p_{\mathbf{v}'} \sim \exp\left(\frac{-\|\mathbf{v}' - \mathbf{t}'\|^2}{2\sigma^2}\right)$$

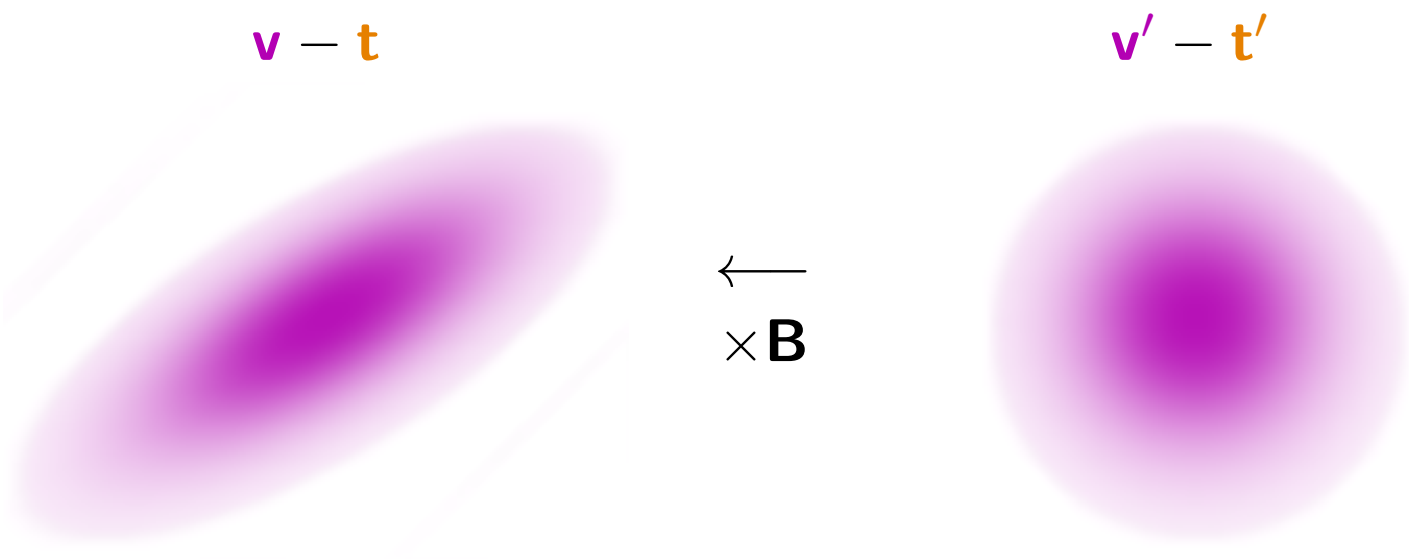
Pour $\sigma \to 0$ on retombe sur l'algorithme non randomisé. Par contre

Théorème (Lissage gaussien [GPV08])

Si $\sigma \geqslant O(\sqrt{\log \frac{1}{\epsilon}})$, pour $\mathbf{t}'$ "uniforme" et $\mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}$ alors la distrib. de $\mathbf{v}' - \mathbf{t}'$ est proche d'une gaussienne continue :

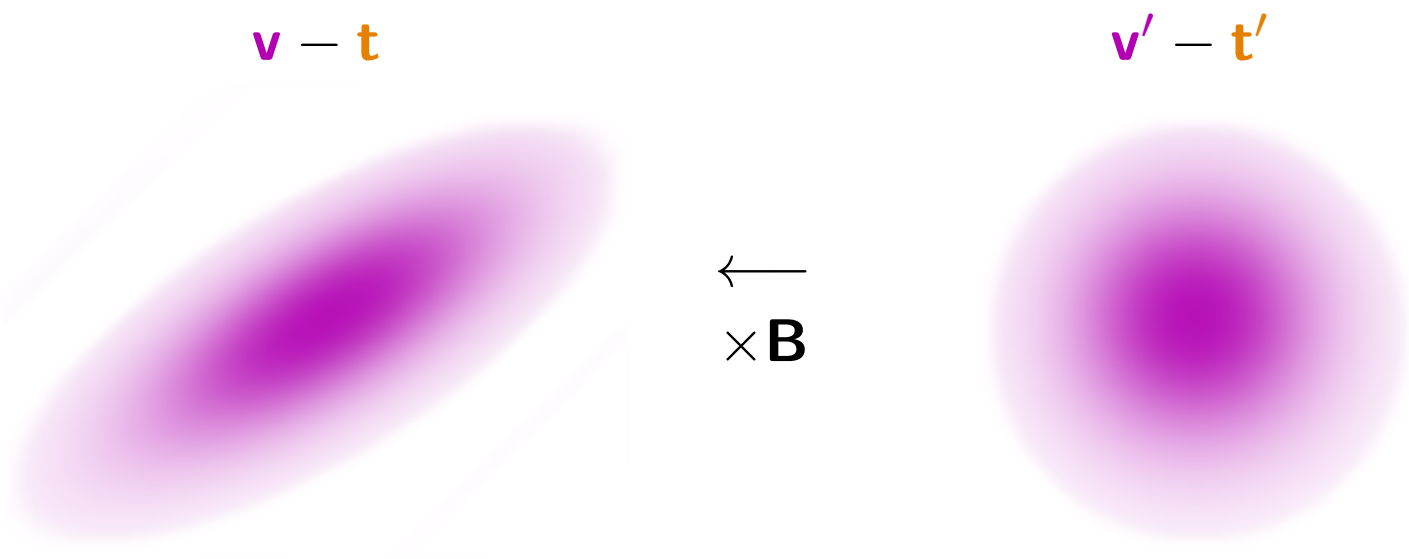
$$\mathbf{v}' - \mathbf{t}' \approx_{n\epsilon} D_{\mathbb{R}^n,\sigma}.$$

Utiliser une base sans la révéler



Avec la randomisation gaussienne, l'algorithme de Babai ne révèle que $\mathrm{Cov}(\mathbf{v} - \mathbf{t}) = \mathbf{B}^t \cdot \mathbf{B}$. On peut éliminer toute dépendance :

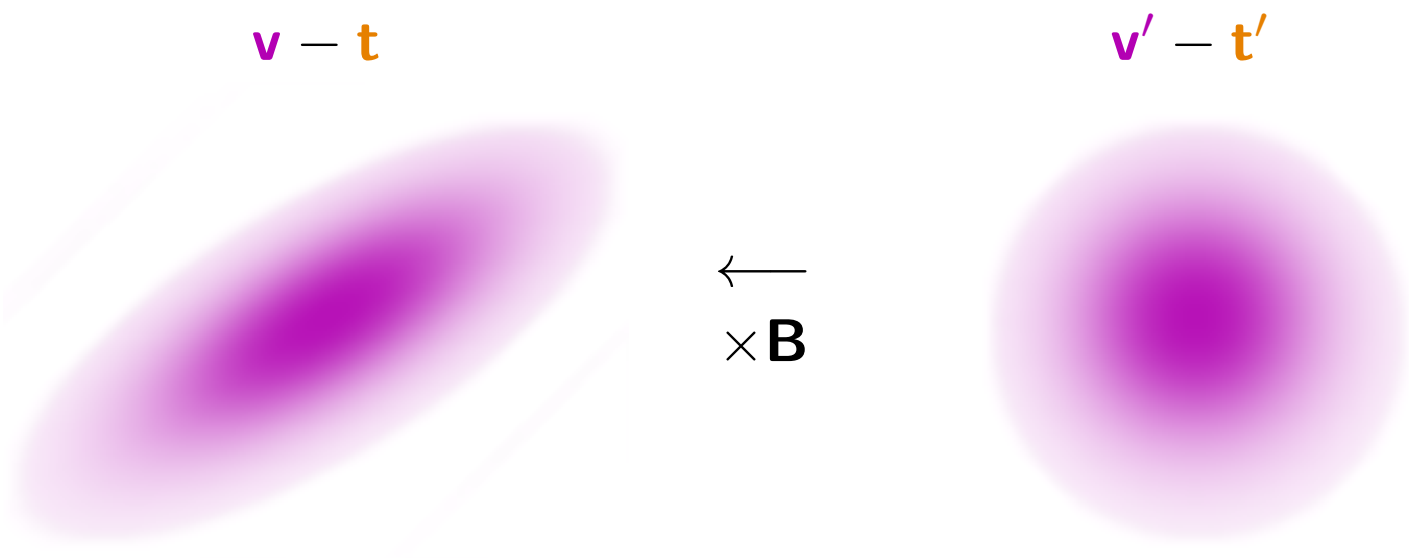
Utiliser une base sans la révéler



Avec la randomisation gaussienne, l'algorithme de Babai ne révèle que $\mathrm{Cov}(\mathbf{v} - \mathbf{t}) = \mathbf{B}^t \cdot \mathbf{B}$. On peut éliminer toute dépendance :

- Orthogonalisation (2^{nd} algorithme de Babai) [Kle00, GPV08]

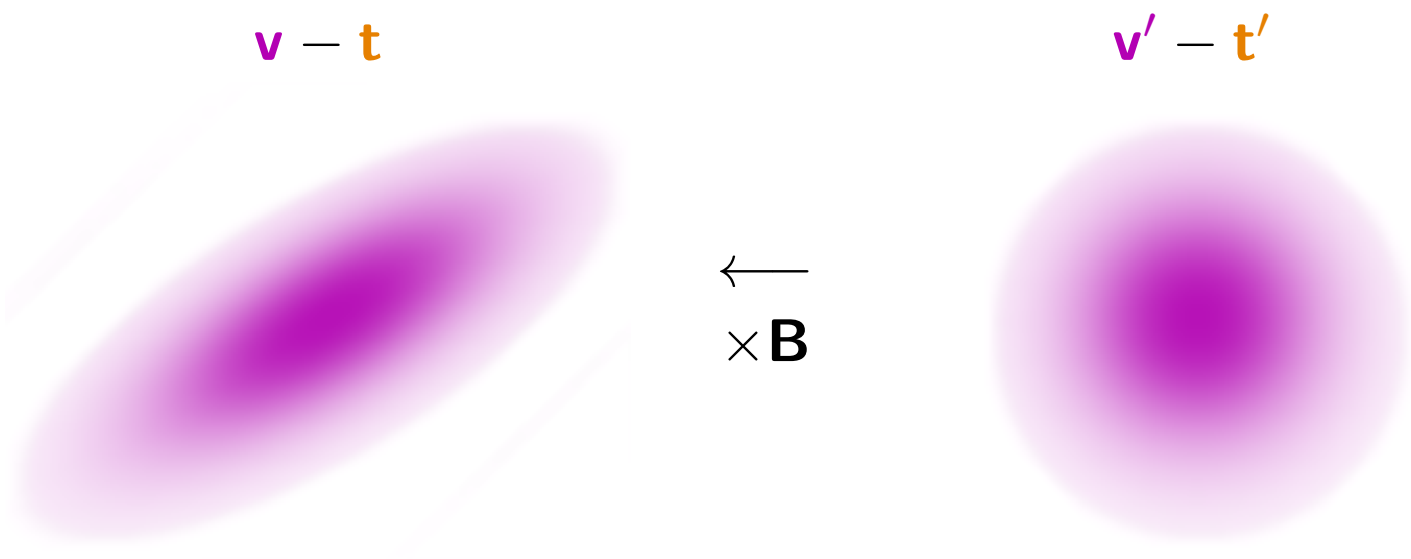
Utiliser une base sans la révéler



Avec la randomisation gaussienne, l'algorithme de Babai ne révèle que $\mathrm{Cov}(\mathbf{v} - \mathbf{t}) = \mathbf{B}^t \cdot \mathbf{B}$. On peut éliminer toute dépendance :

- Orthogonalisation (2^{nd} algorithme de Babai) [Kle00, GPV08]
- Perturbation gaussienne complémentaire [Pei10, MP12]

Utiliser une base sans la révéler



Avec la randomisation gaussienne, l'algorithme de Babai ne révèle que $\mathrm{Cov}(\mathbf{v} - \mathbf{t}) = \mathbf{B}^t \cdot \mathbf{B}$. On peut éliminer toute dépendance :

- Orthogonalisation (2^{nd} algorithme de Babai) [Kle00, GPV08]
- Perturbation gaussienne complémentaire [Pei10, MP12]

Conséquence : Plus de fuite d'information.

$\Rightarrow$ Signature **avec preuve de sécurité** dans le modèle de l'oracle aléatoire.

Efficacité de l'échantillonnage Gaussien Discret (ÉGD)

Les algorithmes de [Kle00, GPV08, Pei10] échantillonnent selon :

$$D_{L,\sigma,\mathbf{t}'}$$

Ils opèrent dans $\mathbb{R} \Rightarrow$ approximation par **l'arithmétique flottante**.

Question : Quelle précision est requise ? Quelle efficacité au final ?

Efficacité de l'échantillonnage Gaussien Discret (ÉGD)

Les algorithmes de [Kle00, GPV08, Pei10] échantillonnent selon :

$$D_{L,\sigma,\mathbf{t}'}$$

Ils opèrent dans $\mathbb{R} \Rightarrow$ approximation par **l'arithmétique flottante**.

Question : Quelle précision est requise ? Quelle efficacité au final ?

Définition (ϵ-correction de l'ÉGD)

Un algorithme d'ÉGD sera dit ϵ-correct si sa distribution de sortie $\mathcal{D}$ est à distance statistique de $D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}$ inférieure à ϵ :

$$\mathcal{D} \approx_\epsilon D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}.$$

Efficacité de l'échantillonnage Gaussien Discret (ÉGD)

Les algorithmes de [Kle00, GPV08, Pei10] échantillonnent selon :

$$D_{L,\sigma,\mathbf{t}'}$$

Ils opèrent dans $\mathbb{R} \Rightarrow$ approximation par **l'arithmétique flottante**.

Question : Quelle précision est requise ? Quelle efficacité au final ?

Définition (ϵ-correction de l'ÉGD)

Un algorithme d'ÉGD sera dit ϵ-correct si sa distribution de sortie $\mathcal{D}$ est à distance statistique de $D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}$ inférieure à ϵ :

$$\mathcal{D} \approx_\epsilon D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}.$$

Pour les applications cryptographiques, on requiert l'ϵ-correction pour $\epsilon = 2^{-\Omega(n)}$.

Notre première analyse

Nous analysons les algorithmes de [Kle00, GPV08, Pei10] :

Théorème (Efficacité de l'ÉGD flottant, [D. Nguyen '12])

Si $\sigma, \mathbf{t} \in \mathbb{Z}^n, \mathbf{B} \in \mathbb{Z}^{n \times n}$ *ont des entrées de taille* poly(n), *les algorithmes sus-cités sont* $2^{-\Omega(n)}$*-corrects pour*

- *une taille de mantisse* $m = O(n)$
- *soit une complexité asymptotique de* $\tilde{O}(n^3)$.

Notre première analyse

Nous analysons les algorithmes de [Kle00, GPV08, Pei10] :

Théorème (Efficacité de l'ÉGD flottant, [D. Nguyen '12])

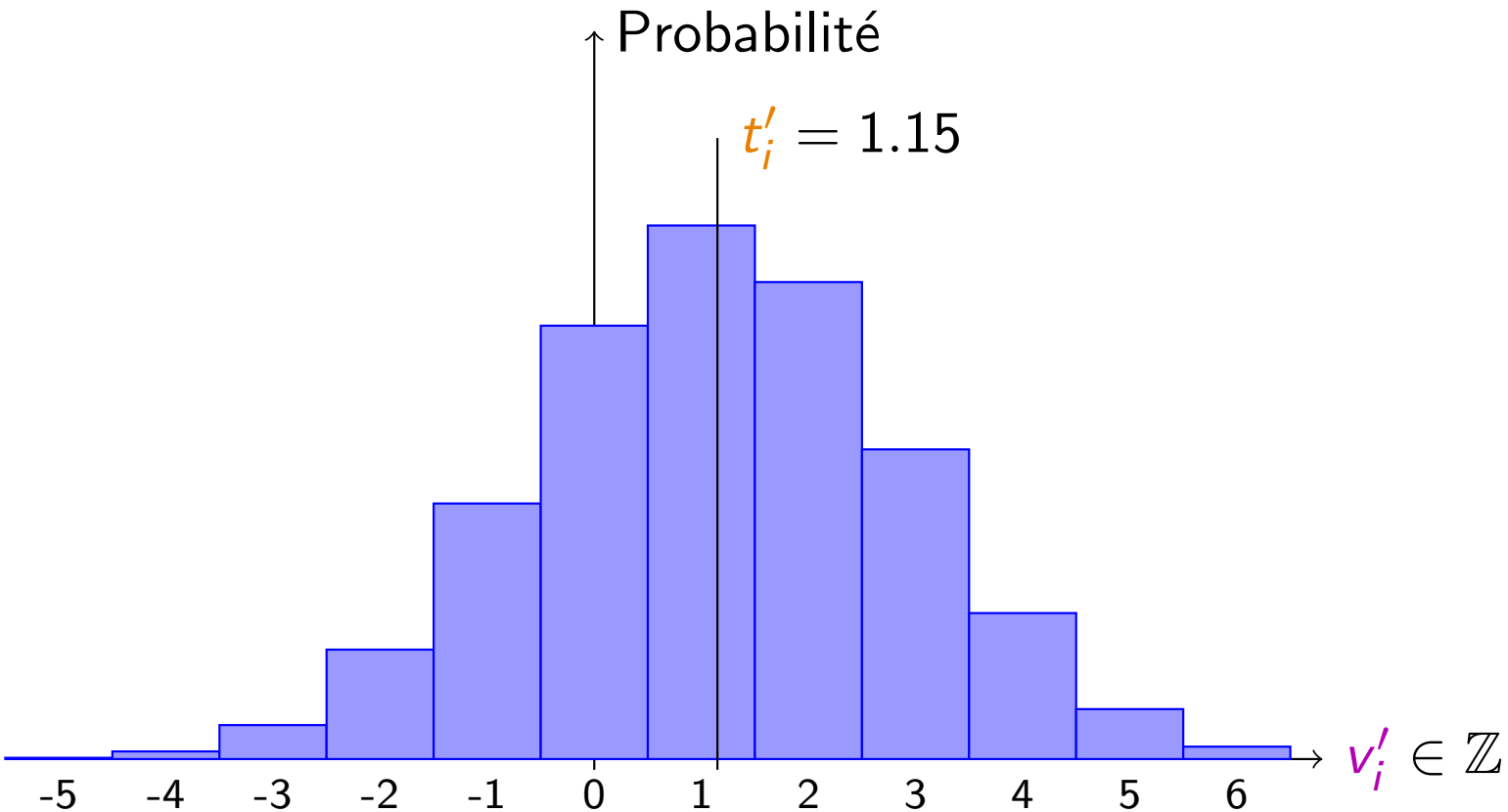
Si $\sigma, \mathbf{t} \in \mathbb{Z}^n, \mathbf{B} \in \mathbb{Z}^{n \times n}$ *ont des entrées de taille* poly(n), *les algorithmes sus-cités sont* $2^{-\Omega(n)}$*-corrects pour*

- *une taille de mantisse* $m = O(n)$
- *soit une complexité asymptotique de* $\tilde{O}(n^3)$.

Peu satisfaisant : On souhaiterait $\tilde{O}(n^2)$ en général voir $\tilde{O}(n)$ pour les réseaux structurés. **Optimisations ?**

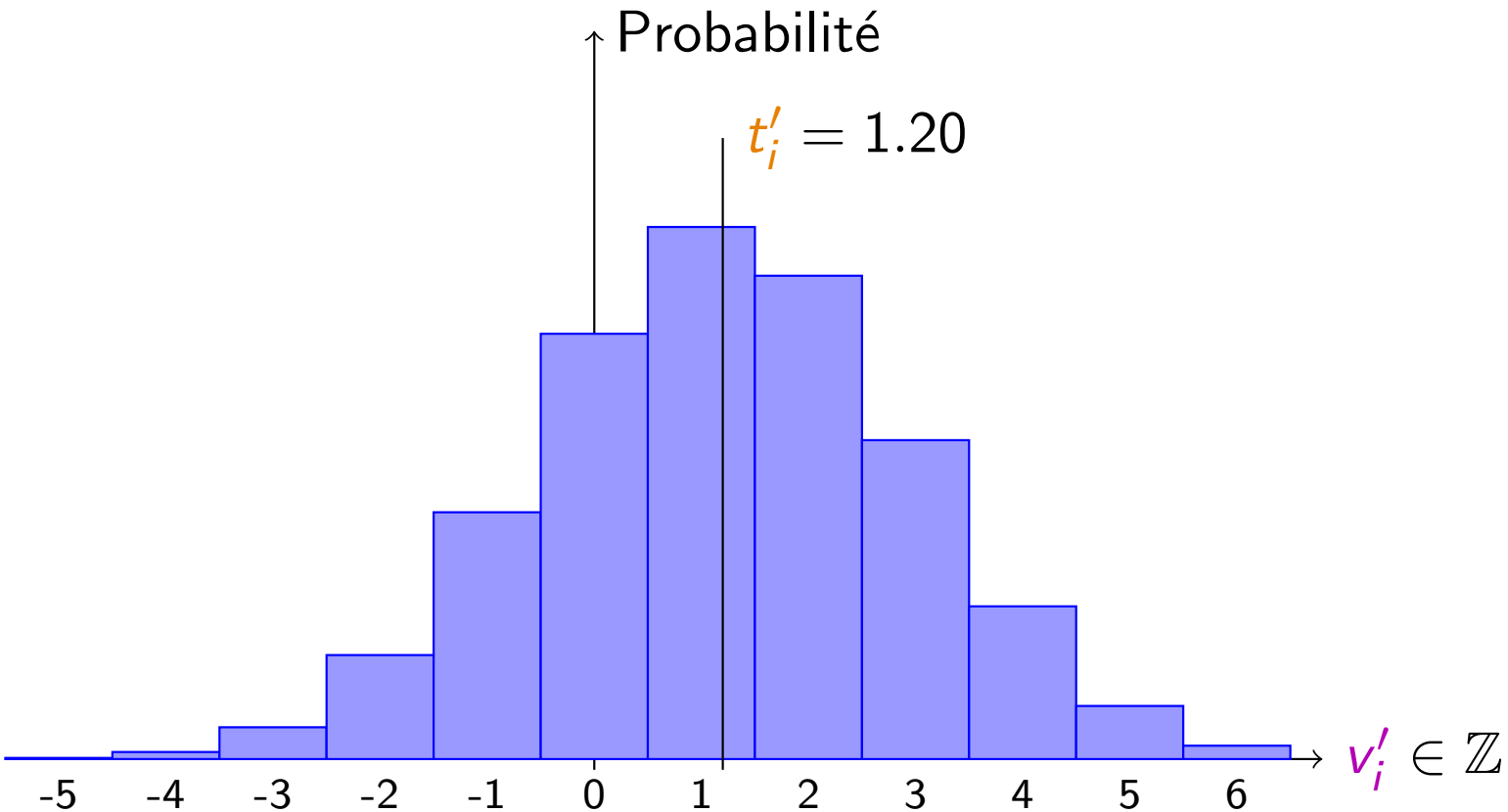
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



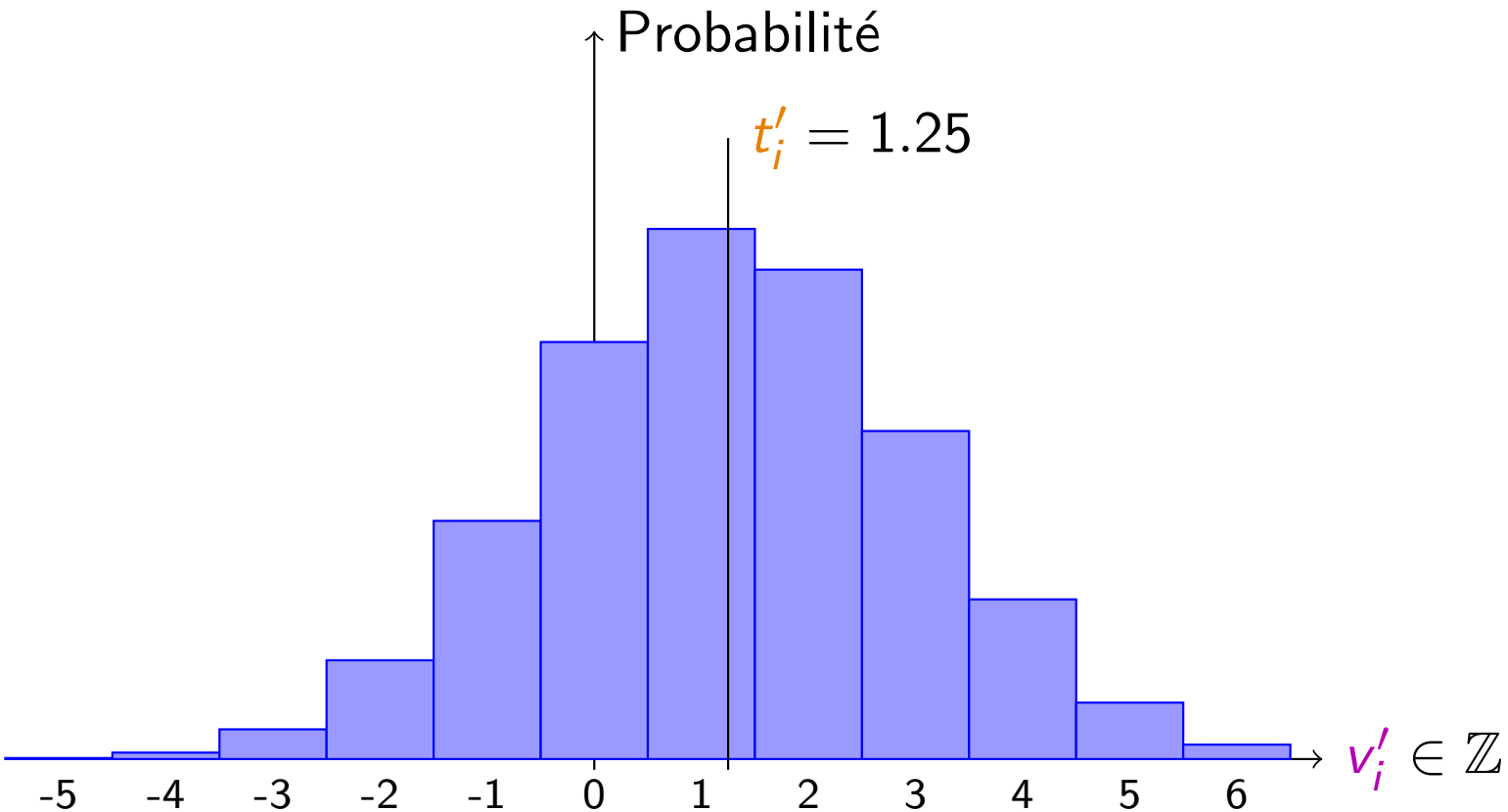
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



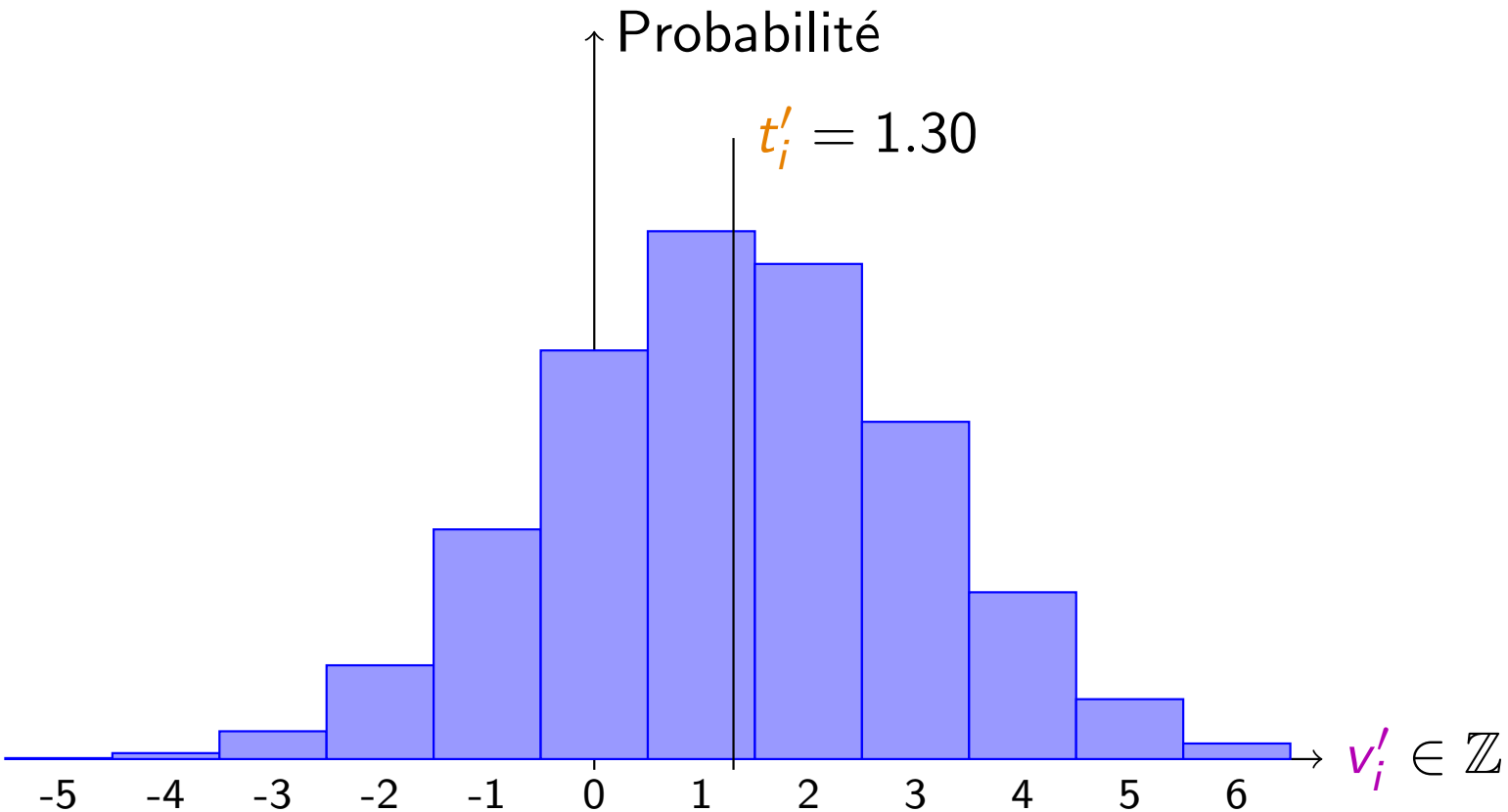
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



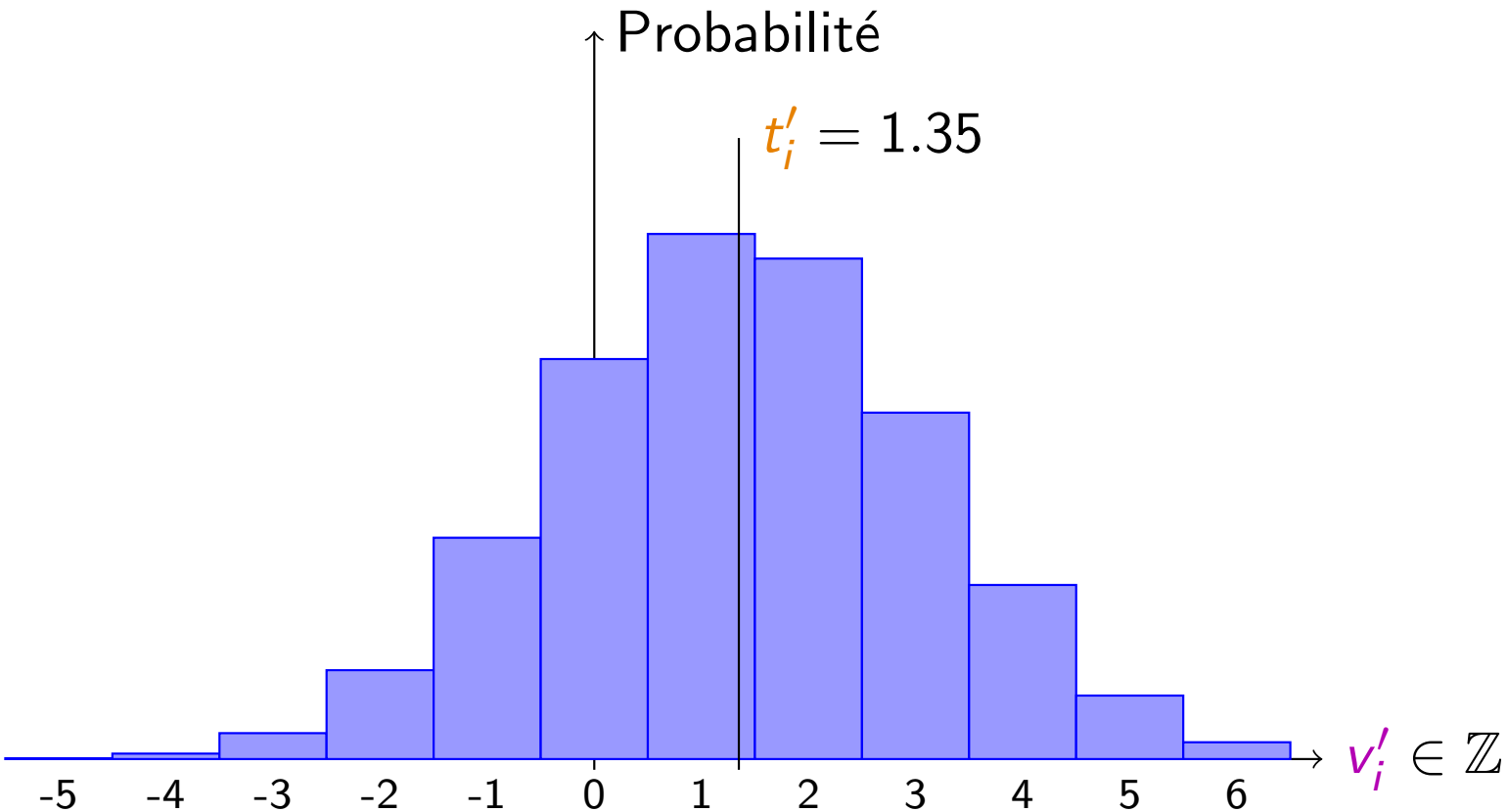
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



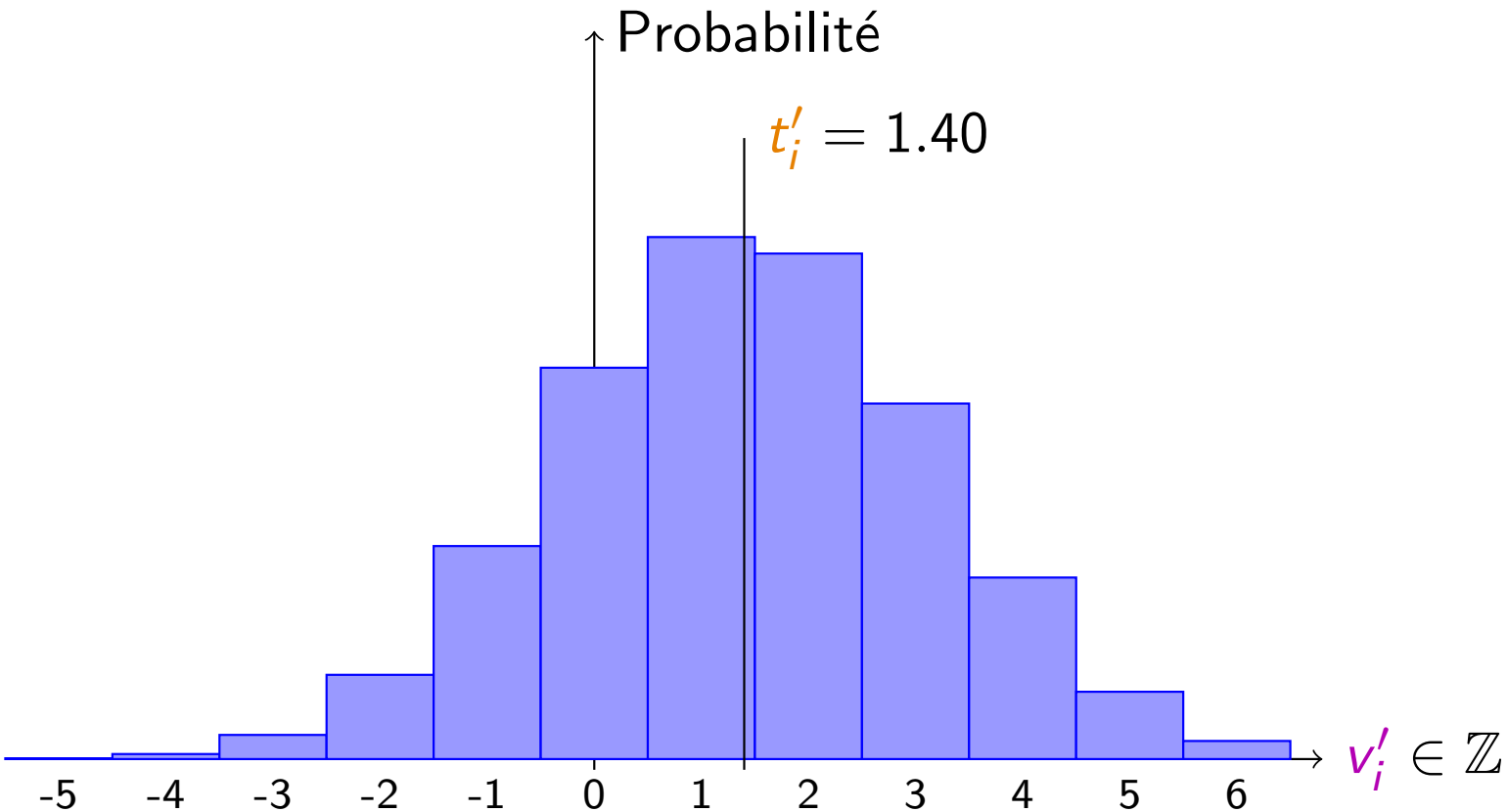
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



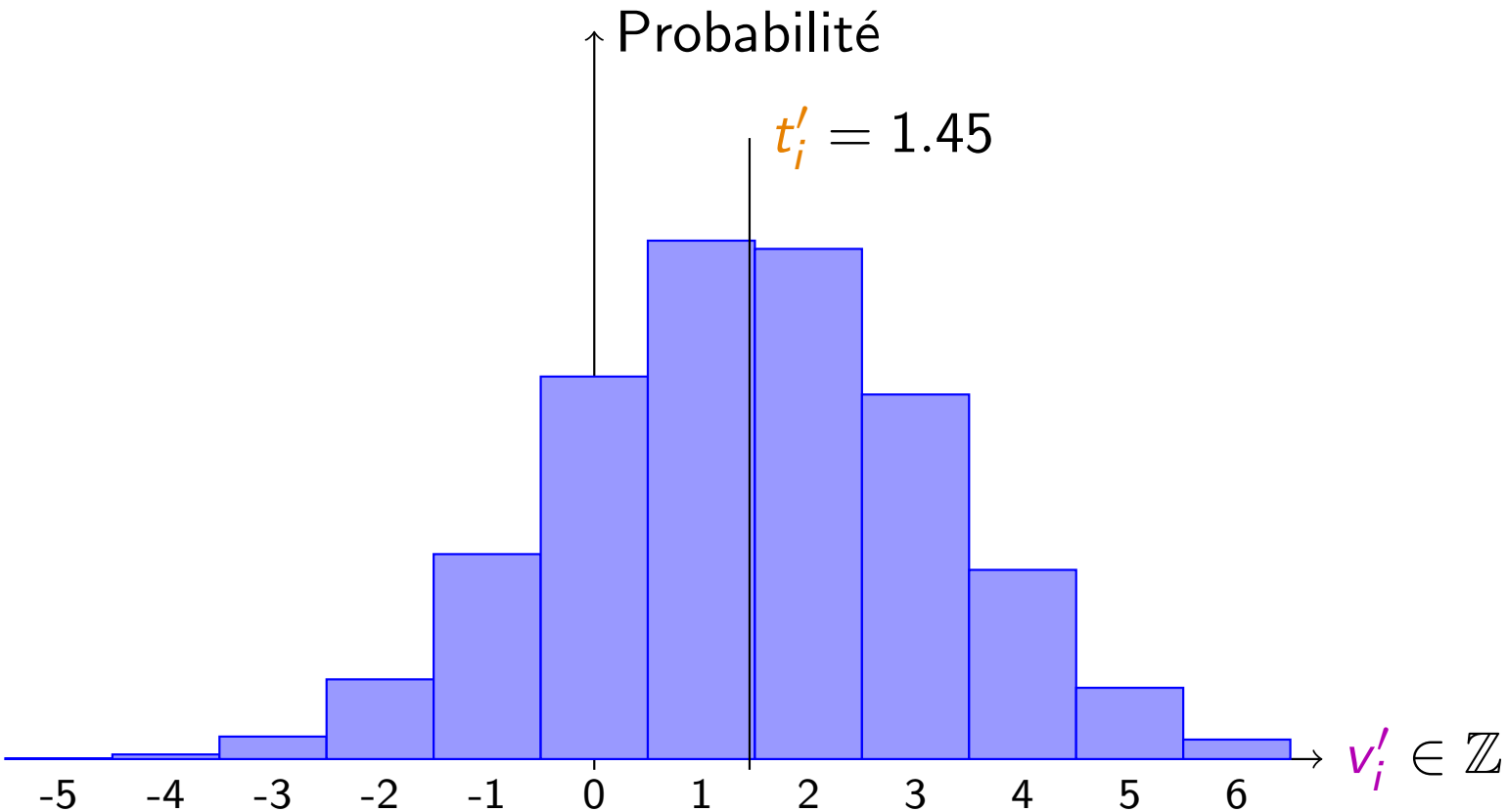
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



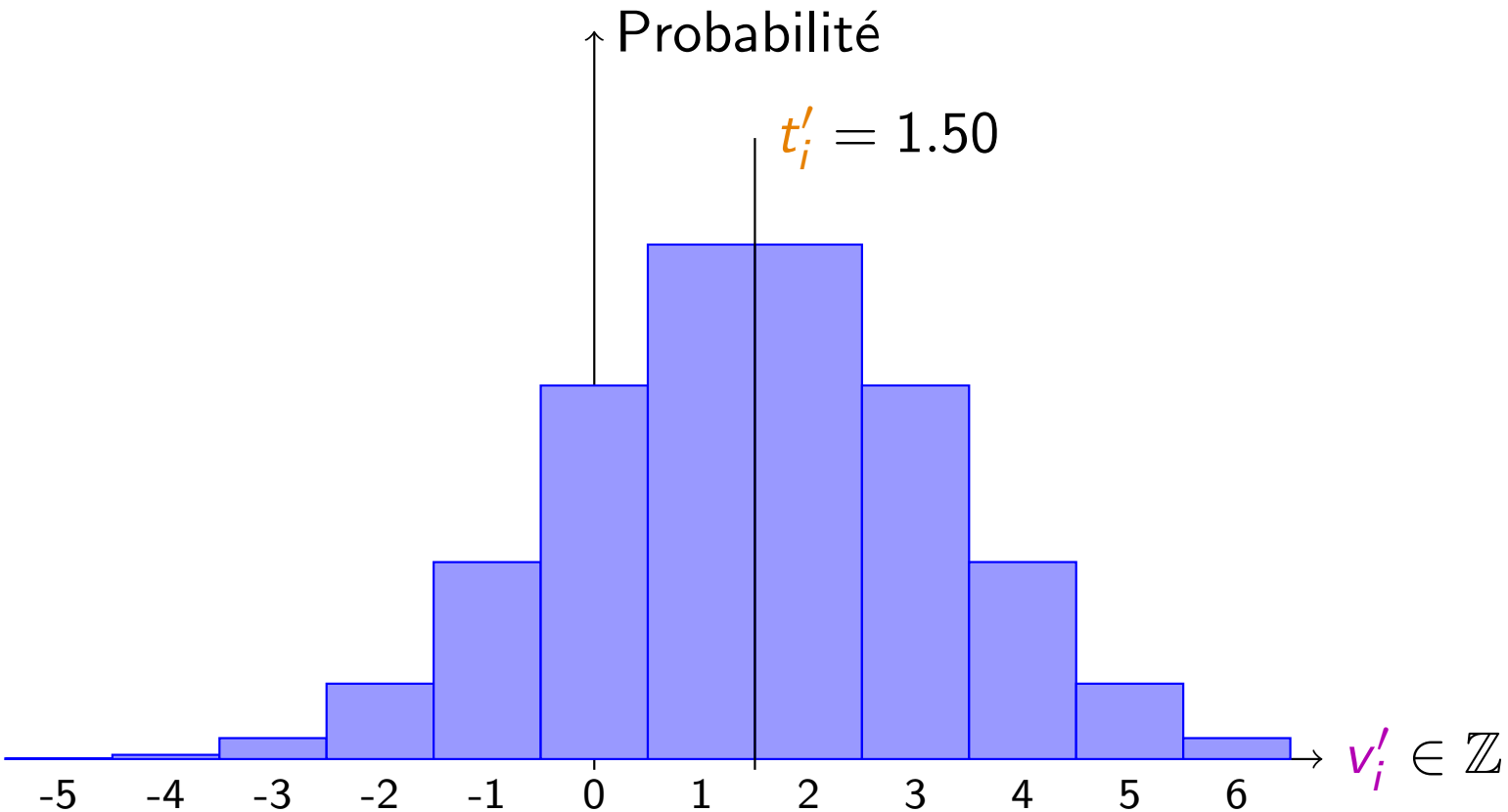
ÉGD en dimension 1 : Le point clef

$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



ÉGD en dimension 1 : Le point clef

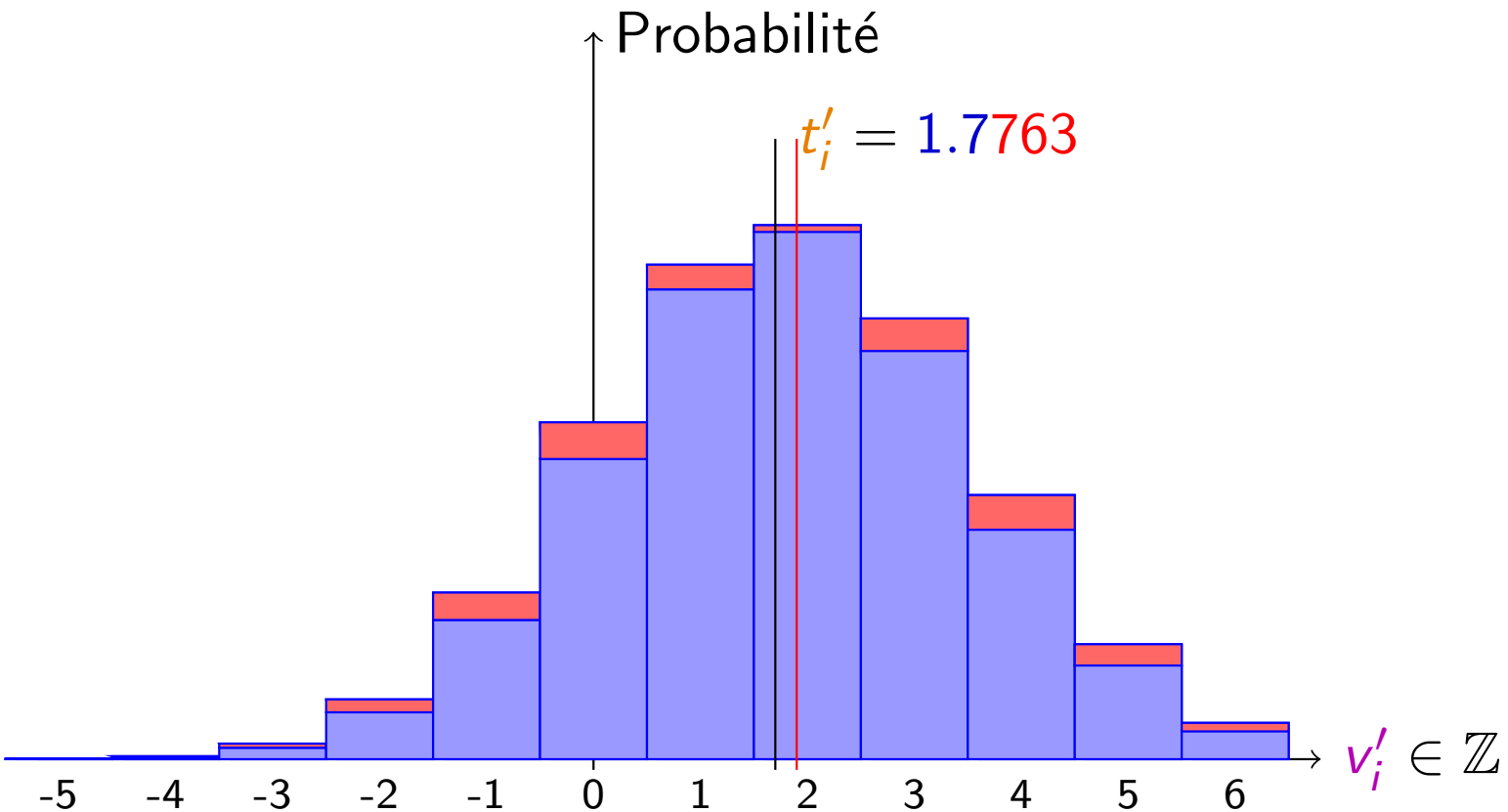
$$\mathbf{t}' = \mathbf{B}^{-1} \cdot \mathbf{t}; \qquad \mathbf{v}' \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{t}'}; \qquad \mathbf{v} = \mathbf{B} \cdot \mathbf{v}'$$



ÉGD en dimension 1 : Le point clef

Propagation de l'incertitude

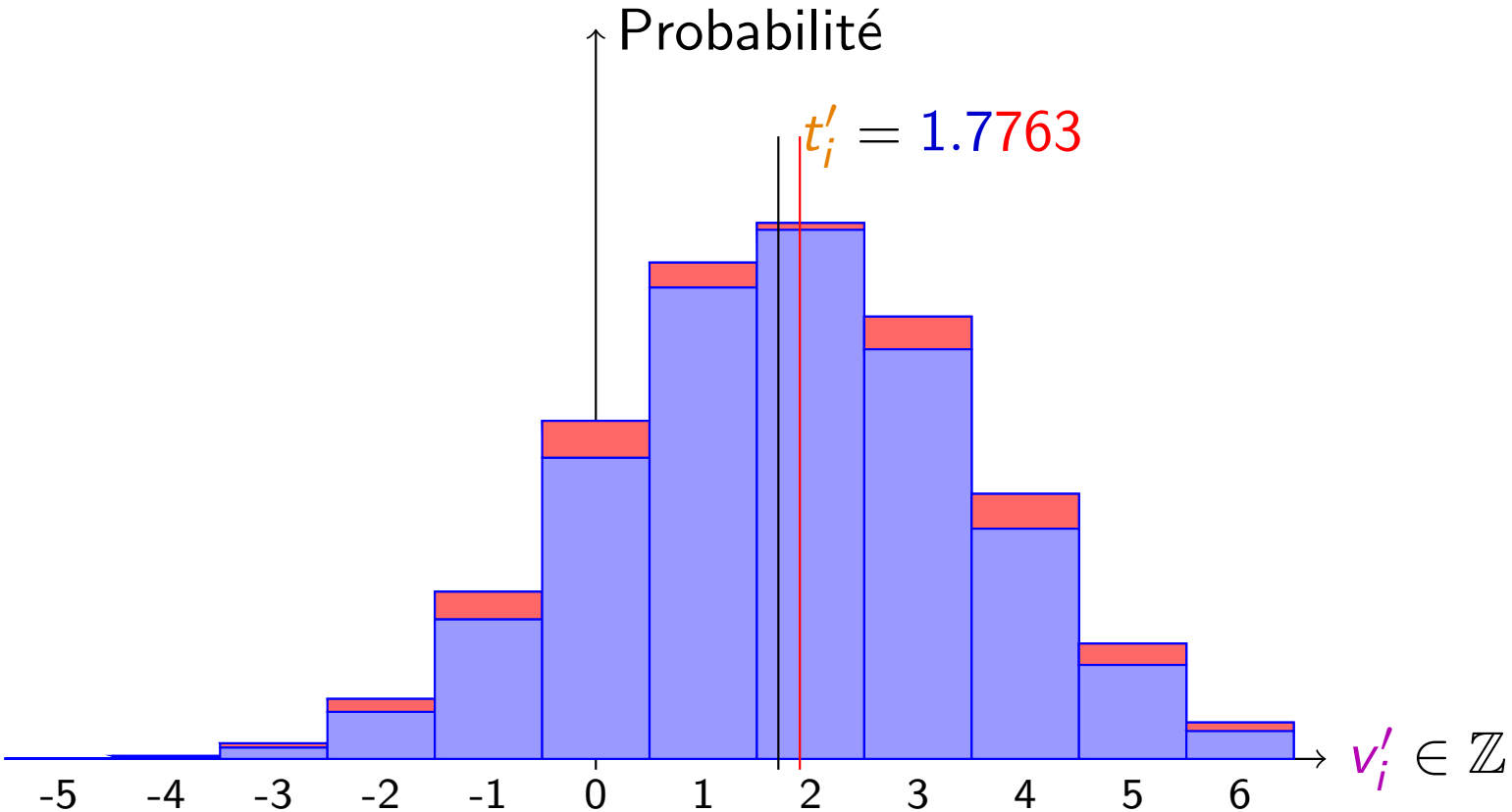
La précision de la distribution échantillonnée dépend de la précision de calcul du centre c.



ÉGD en dimension 1 : Le point clef

Condition de correction

On souhaite rendre l'incertitude négligeable ($2^{-\Omega(n)}$).

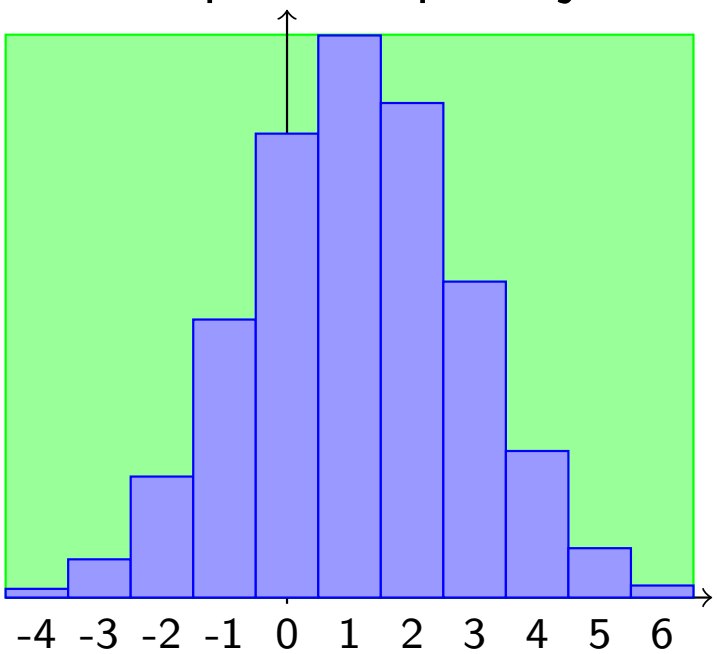


ÉGD en dimension 1 : tirage avec rejet (méthode standard)

Pour échantillonner la gaussienne discrète on procède par rejet.

Tirage avec rejet :

- Échantillonner $(x, y) \in$ ■
- Si $(x, y) \in$ ■ retourner x
- Sinon, recommencer

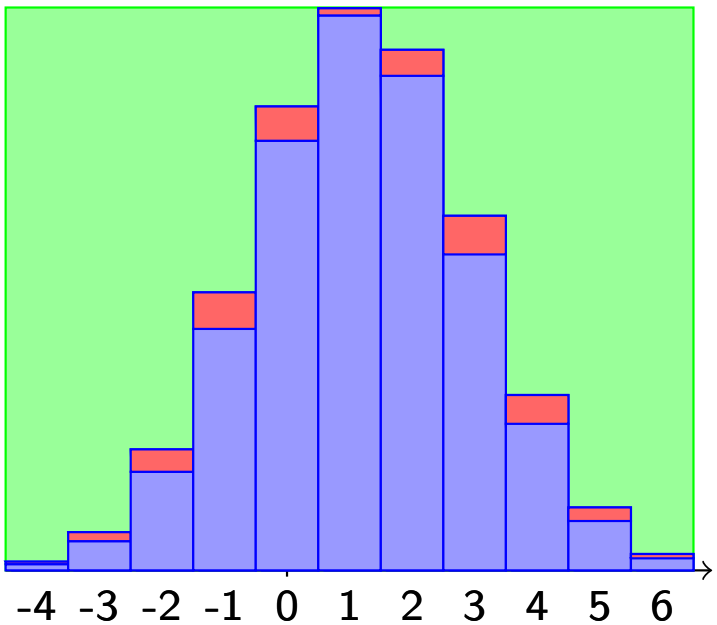


ÉGD en dimension 1 : gérer l'incertitude (notre approche)

L'incertitude n'influe pas toujours sur le résultat final.
Gérer ces cas à part avec une **alerte**.

Tirage avec rejet et alerte :

- Échantillonner $(x, y) \in$ ■
- Si $(x, y) \in$ ■ **alerte**
- Si $(x, y) \in$ ■ renvoyer x
- Sinon, recommencer



Optimisation : algorithmes paresseux

Utiliser deux précisions : **haute prec.** m and **basse prec.** $m' < m$.

Haute prec. $\Rightarrow$ zone ■ négligeable : $2^{-\Theta(n)}$

Basse prec. $\Rightarrow$ zone ■ petite : $O(1/n^{\Theta(1)})$ (alertes rares)

Échantillonnage **paresseux** avec rejet :

- Calculer t'_i en basse précision
- Lancer le tirage avec rejet et alertes

Optimisation : algorithmes paresseux

Utiliser deux précisions : **haute prec.** m and **basse prec.** $m' < m$.

Haute prec. $\Rightarrow$ zone ■ négligeable : $2^{-\Theta(n)}$

Basse prec. $\Rightarrow$ zone ■ petite : $O(1/n^{\Theta(1)})$ (alertes rares)

Échantillonnage **paresseux** avec rejet :

- Calculer t_i' en basse précision
- Lancer le tirage avec rejet et alertes
- Si l'alerte est déclenchée

Optimisation : algorithmes paresseux

Utiliser deux précisions : **haute prec.** m and **basse prec.** $m' < m$.

Haute prec. $\Rightarrow$ zone ■ négligeable : $2^{-\Theta(n)}$

Basse prec. $\Rightarrow$ zone ■ petite : $O(1/n^{\Theta(1)})$ (alertes rares)

Échantillonnage **paresseux** avec rejet :

- Calculer t'_i en basse précision
- Lancer le tirage avec rejet et alertes
- Si l'alerte est déclenchée

  Recalculer t'_i en haute précision

Optimisation : algorithmes paresseux

Utiliser deux précisions : **haute prec.** m and **basse prec.** $m' < m$.

Haute prec. $\Rightarrow$ zone ■ négligeable : $2^{-\Theta(n)}$

Basse prec. $\Rightarrow$ zone ■ petite : $O(1/n^{\Theta(1)})$ (alertes rares)

Échantillonnage **paresseux** avec rejet :

- Calculer t'_i en basse précision
- Lancer le tirage avec rejet et alertes
- Si l'alerte est déclenchée

  Recalculer t'_i en haute précision

  Reprendre le tirage, avec zone d'incertitude ■ négligeable

Efficacité de nos algorithmes paresseux

Nos algorithmes paresseux atteignent la **complexité souhaitée** :

Théorème (Efficacité de l'ÉGD flottant et paresseux, [D. Nguyen '12])

Si $\sigma, \mathbf{t} \in \mathbb{Z}^n, \mathbf{B} \in \mathbb{Z}^{n \times n}$ *et* $\mathbf{B}^{-1}$ *ont des entrée de taille* $\mathrm{poly}(n)$, *nos algorithmes paresseux sont* $O(2^{-n})$*-corrects pour :*

- *des tailles de mantisse* $m = O(n)$ *et* $m' = \mathrm{polylog}(n)$
- *et sa complexité asymptotique est de* $\tilde{O}(n^2)$

Si de plus la base $\mathbf{B}$ *est structurée, certains ont une complexité quasi-linéaire* $\tilde{O}(n)$.

De plus nos résultats sont **concrets** : constantes des $\tilde{O}$ explicitées
$\Rightarrow$ résultats directement **applicables en pratique**.

Échantillonnage sans arithmétique flottante ?

L'arithmétique flottante, même à basse précision est un problème sur **petite architecture** (ex. carte-à-puce).

Question : Est-elle vraiment nécessaire ?

Échantillonnage sans arithmétique flottante ?

L'arithmétique flottante, même à basse précision est un problème sur **petite architecture** (ex. carte-à-puce).

Question : Est-elle vraiment nécessaire ?

- **Non** ! au moins pour des cas simples :
  gaussienne centrée en $\mathbf{0}$ sur $\mathbb{Z}^n$ (application : BLISS)

Échantillonnage sans arithmétique flottante ?

L'arithmétique flottante, même à basse précision est un problème sur **petite architecture** (ex. carte-à-puce).

Question : Est-elle vraiment nécessaire ?

- **Non** ! au moins pour des cas simples :
  gaussienne centrée en $\mathbf{0}$ sur $\mathbb{Z}^n$ (application : BLISS)
- et peux-être plus généralement (cf. mémoire)

Algèbre sur les distributions de Bernoulli

On cherche a échantillonner $B_{\exp(-x/f)}$ efficacement
Idée : pré-calculer quelques biais c et combiner les variables de Bernoulli $\mathcal{B}_c$:

Algèbre sur les distributions de Bernoulli

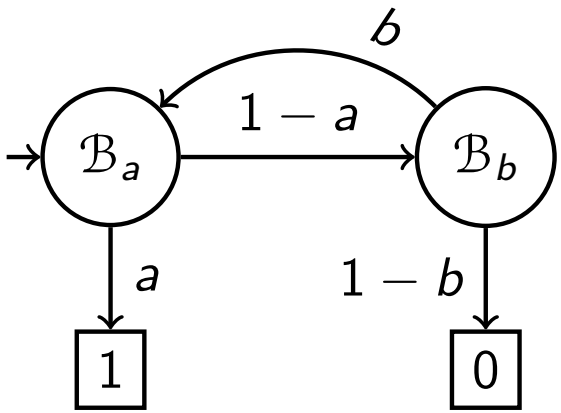
On cherche a échantillonner $B_{\exp(-x/f)}$ efficacement
Idée : pré-calculer quelques biais c et combiner les variables de Bernoulli $\mathcal{B}_c$:

- $\mathcal{B}_a \wedge \mathcal{B}_b = \mathcal{B}_{ab}$
- $\mathcal{B}_a \vee \mathcal{B}_b = \mathcal{B}_{a+b-ab}$
- $\mathcal{B}_a \oplus \mathcal{B}_b = \mathcal{B}_{a+b-2ab}$

Algèbre sur les distributions de Bernoulli

On cherche a échantillonner $B_{\exp(-x/f)}$ efficacement
Idée : pré-calculer quelques biais c et combiner les variables de Bernoulli $\mathcal{B}_c$:

- $\mathcal{B}_a \wedge \mathcal{B}_b = \mathcal{B}_{ab}$
- $\mathcal{B}_a \vee \mathcal{B}_b = \mathcal{B}_{a+b-ab}$
- $\mathcal{B}_a \oplus \mathcal{B}_b = \mathcal{B}_{a+b-2ab}$
- $\mathcal{B}_a \oslash \mathcal{B}_b = \mathcal{B}_{a/(1-(1-a)b)}$

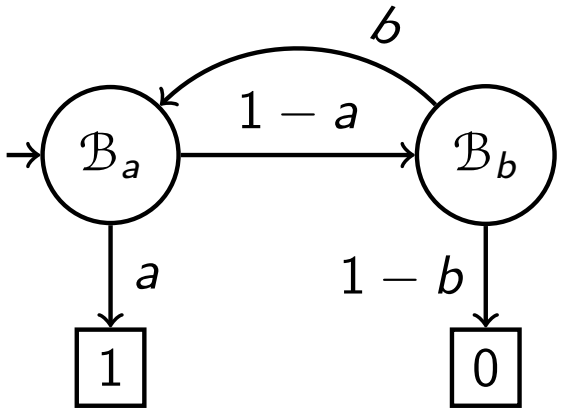


Calcul de $\mathcal{B}_a \oslash \mathcal{B}_b$

Algèbre sur les distributions de Bernoulli

On cherche a échantillonner $B_{\exp(-x/f)}$ efficacement
Idée : pré-calculer quelques biais c et combiner les variables de Bernoulli $\mathcal{B}_c$:

- $\mathcal{B}_a \wedge \mathcal{B}_b = \mathcal{B}_{ab}$
- $\mathcal{B}_a \vee \mathcal{B}_b = \mathcal{B}_{a+b-ab}$
- $\mathcal{B}_a \oplus \mathcal{B}_b = \mathcal{B}_{a+b-2ab}$
- $\mathcal{B}_a \oslash \mathcal{B}_b = \mathcal{B}_{a/(1-(1-a)b)}$



Calcul de $\mathcal{B}_a \oslash \mathcal{B}_b$

Algorithmes sans flottants ni fonctions transcendantes

Algorithmes pour $\mathcal{B}_{\exp(-x/f)}$ et $\mathcal{B}_{1/\cosh(x/f)}$, pour tout $x \in \mathbb{Z} \cap [0, B]$ efficacement avec seulement $\log B$ constantes pré-calculées.

Plan

Introduction

1 Signatures numériques

2 Réseaux euclidiens : géométrie et algorithmes

Contributions

3 Signatures heuristiques et attaques

4 Analyse et optimisation de l'ÉGD

5 BLISS : Un schéma de signature efficace

Conclusion

6 Conclusion et perspectives

Signature à la Fiat-Shamir fondée sur les réseaux

Les constructions précédentes s'appuient sur l'algorithme de Babai (randomisé) pour utiliser une base courte comme une **trappe**.

Alternative : paradigme de Fiat-Shamir [Lyu09, Lyu12, GLP12]

- **sans trappe** $\Rightarrow$ ÉGD seulement sur $\mathbb{Z}^n$
- avec preuve de sécurité
- état de l'art compétitif : signature de 10kb pour 80 bits de sécurité.

Objectif : concevoir un schéma avec sécurité prouvable et vraiment compétitif.

Paradigme de Fiat-Shamir

Protocole d'identification en 3 tours :

Théorème (Transformation générique de Fiat-Shamir [AABN02])

Schéma d'identification en 3 tours sûr contre adversaire passif
$\rightarrow$ *Schéma de signature sûr dans le modèle de l'oracle aléatoire.*

Fiat-Shamir sur les réseaux, premier essai

sk : $\mathbf{S} \in \mathbb{Z}_q^{m \times k}$, petite

pk : $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, aléatoire
$\mathbf{T} = \mathbf{AS}$ (pseudo-)aléatoire

Choisir $\mathbf{y} \in \mathbb{Z}^m$, court $\xrightarrow{\mathbf{w} = \mathbf{Ay} \in \mathbb{Z}^n}$

Problème : Chaque signature $(\mathbf{z}, \mathbf{c})$ laisse fuir de l'information sur la clef secrète $\mathbf{S}$.

Fiat-Shamir sur les réseaux, premier essai

sk : $\mathbf{S} \in \mathbb{Z}_q^{m \times k}$, petite

pk : $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, aléatoire
$\mathbf{T} = \mathbf{AS}$ (pseudo-)aléatoire

Choisir $\mathbf{y} \in \mathbb{Z}^m$, court

$\mathbf{w} = \mathbf{Ay} \in \mathbb{Z}^n$ →

Choisir $\mathbf{c} \in \mathbb{Z}^k$,court

← $\mathbf{c}$

Problème : Chaque signature $(\mathbf{z}, \mathbf{c})$ laisse fuir de l'information sur la clef secrète $\mathbf{S}$.

Fiat-Shamir sur les réseaux, premier essai

sk : $\mathbf{S} \in \mathbb{Z}_q^{m \times k}$, petite

pk : $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, aléatoire
$\mathbf{T} = \mathbf{AS}$ (pseudo-)aléatoire

Choisir $\mathbf{y} \in \mathbb{Z}^m$, court

$\mathbf{w} = \mathbf{Ay} \in \mathbb{Z}^n$ →

Choisir $\mathbf{c} \in \mathbb{Z}^k$,court

← $\mathbf{c}$

$\mathbf{z} = \mathbf{y} + \mathbf{Sc}$

$\mathbf{z}$ →

Verifier que
1) $\|\mathbf{z}\|$ est petit, et
2) $\mathbf{Az} = \mathbf{Tc} + \mathbf{w} \bmod q$

Problème : Chaque signature $(\mathbf{z}, \mathbf{c})$ laisse fuir de l'information sur la clef secrète $\mathbf{S}$.

Fiat-Shamir avec abandon [Lyu12]

sk : $\mathbf{S} \in \mathbb{Z}_q^{m \times k}$, petite

pk : $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, aléatoire
$\mathbf{T} = \mathbf{AS}$ (pseudo-)aléatoire

Choisir $\mathbf{y} \leftarrow D_{\mathbb{Z},\sigma}^m$, court

$\mathbf{w} = \mathbf{Ay} \in \mathbb{Z}^n$ →

Choisir $\mathbf{c} \in \mathbb{Z}^k$,court

← $\mathbf{c}$

$\mathbf{z} = \mathbf{y} + \mathbf{Sc}$

Abandonner avec probabilité
$1 - \frac{\exp(-\|\mathbf{x}\|^2/2\sigma^2)}{M \exp(-\|\mathbf{x}-\mathbf{Sc}\|^2/2\sigma^2)}$

$\mathbf{z}$ →

Vérifier que
1) $\|\mathbf{z}\|$ est petit, et
2) $\mathbf{Az} = \mathbf{Tc} + \mathbf{w} \bmod q$

La probabilité d'abandon est telle que $\mathbf{z} \sim D_{\mathbb{Z},\sigma}^m$ soit indép. de $\mathbf{S}$
$\Rightarrow$ **Plus de fuite d'information !**

Notre optimisation principale : Gaussienne bi-modale

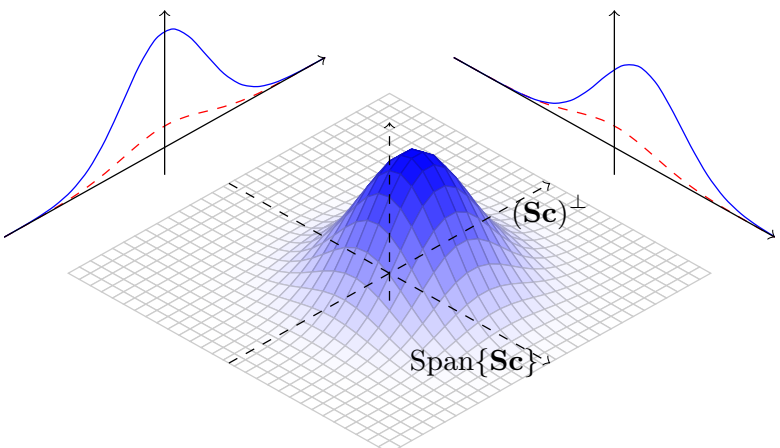
Notre analyse montre que, si l'on pouvait remplacer

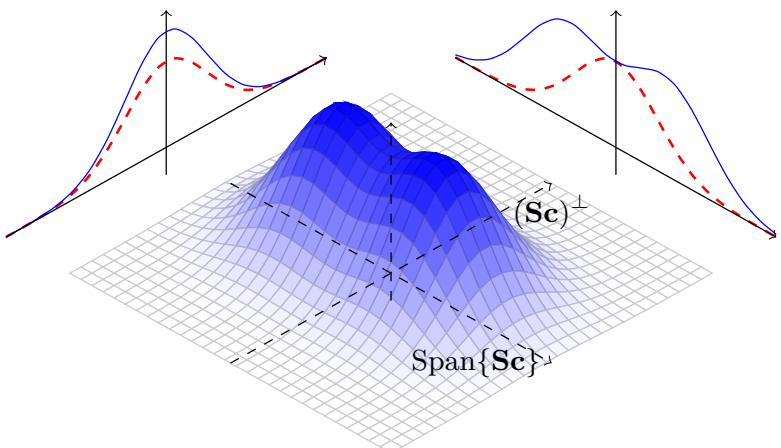
$$\mathbf{z} = \mathbf{y} + \mathbf{Sc} \text{ par } \mathbf{z} = \mathbf{y} \pm \mathbf{Sc}$$

alors la probabilité d'abandon baisserait drastiquement.

$\Rightarrow$ réduire σ d'un facteur $O(\sqrt{\lambda})$ (de 12 à 24 en pratique)

$\Rightarrow$ Signatures **plus courtes** et plus difficile **à falsifier** !





Notre optimisation principale : Gaussienne bi-modale

Notre analyse montre que, si l'on pouvait remplacer

$$\mathbf{z} = \mathbf{y} + \mathbf{Sc} \text{ par } \mathbf{z} = \mathbf{y} \pm \mathbf{Sc}$$

alors la probabilité d'abandon baisserait drastiquement.

$\Rightarrow$ réduire σ d'un facteur $O(\sqrt{\lambda})$ (de 12 à 24 en pratique)

$\Rightarrow$ Signatures **plus courtes** et plus difficile **à falsifier** !

Pour préserver la sécurité, cela nécessite que $\mathbf{T} = \mathbf{AS} = -\mathbf{T} \neq \mathbf{0}$

Notre optimisation principale : Gaussienne bi-modale

Notre analyse montre que, si l'on pouvait remplacer

$$\mathbf{z} = \mathbf{y} + \mathbf{Sc} \text{ par } \mathbf{z} = \mathbf{y} \pm \mathbf{Sc}$$

alors la probabilité d'abandon baisserait drastiquement.

$\Rightarrow$ réduire σ d'un facteur $O(\sqrt{\lambda})$ (de 12 à 24 en pratique)

$\Rightarrow$ Signatures **plus courtes** et plus difficile **à falsifier** !

Pour préserver la sécurité, cela nécessite que $\mathbf{T} = \mathbf{AS} = -\mathbf{T} \neq \mathbf{0}$

- il est possible de construire de telles clefs mod $2q$
- une construction très efficace ($\mathbf{S}$ très courte) est possible en s'inspirant de NTRUSIGN ;
- étonnamment, la nouvelle **preuve de sécurité** est plus simple et moins lâche !

Autres optimisations

Nous proposons de nombreuses optimisations, nouvelles ou adaptées.

Taille des signatures :

- éliminations de bits peu significatifs [GLP12]
- borne ad-hoc pour le produit **Sc**
- génération de **c** optimale (entropie / longueur)
- compression de Huffmann de la distribution gaussienne.

Efficacité des algorithmes :

- choix de n et q pour un produit rapide dans $\mathbb{Z}_q[X]/(\Phi_n(X))$;
- utilisation des algorithmes d'ÉGD sans flottants.

BLISS : **B**imodal **L**attice **S**ignature **S**cheme

Nous proposons une signature avec paramètres concrets.

BLISS (D. Durmus Lepoint Lyubashevsky '13)

Schéma	**BLISS-0**	**BLISS-I**	**BLISS-II**	**BLISS-III**	**BLISS-IV**
Securité	Jouet	128 bits	128 bits	160 bits	192 bits
Optimisé pour	Fun	Vitesse	Taille	Sécurité	Sécurité
dimension n	256	512	512	512	512
Module q	7681	12289	12289	12289	12289
Répétition	7.4	1.6	7.4	2.8	5.2
Signature	**3.3kb**	**5.6kb**	**5kb**	**6kb**	**6.5kb**
Clef Secrète	1.5kb	2kb	2kb	3kb	3kb
Clef Publique	3.3kb	7kb	7kb	7kb	7kb

Nos niveaux de sécurité affichés considèrent les meilleures attaques connues sur les réseaux généraux et ceux de NTRUSIGN : [GNR10, CN11, HG07, CN11, MM11].

Efficacité de BLISS comparée à RSA et ECDSA

Notre implémentation **open source** se compare favorablement aux cryptosystèmes standards (`OpenSSL` sur `x86-64`).

Implem.	Securité	Sign. (ms)	Sign./s	Ver. (ms)	Ver./s
Toy	$\leqslant$ 60 bits	0.241	4k	0.017	59k
BLISS-I	**128 bits**	**0.124**	**8k**	**0.030**	**33k**
BLISS-II	128 bits	0.480	2k	0.030	33k
BLISS-III	160 bits	0.203	5k	0.031	32k
BLISS-IV	192 bits	0.375	2.5k	0.032	31k
RSA 1024	72-80 bits	0.167	6k	0.004	91k
RSA 2048	112 bits	1.180	0.8k	0.038	27k
RSA 4096	$\geqslant$ **128 bits**	**8.660**	**0.1k**	**0.138**	**7.5k**
ECDSA 160	80 bits	0.058	17k	0.205	5k
ECDSA 256	**128 bits**	**0.106**	**9.5k**	**0.384**	**2.5k**
ECDSA 384	192 bits	0.195	5k	0.853	1k

La comparaison est probablement encore plus favorable sur petite architecture.

Plan

Introduction

1 Signatures numériques

2 Réseaux euclidiens : géométrie et algorithmes

Contributions

3 Signatures heuristiques et attaques

4 Analyse et optimisation de l'ÉGD

5 BLISS : Un schéma de signature efficace

Conclusion

6 Conclusion et perspectives

Publications

- *Learning a Zonotope and More : Cryptanalysis of NTRUSign Countermeasures.*
  D. et P. Nguyen. **ASIACRYPT '12**
- *Faster Gaussian Lattice Sampling using Lazy Floating-Point Arithmetic*
  D. et P. Nguyen. **ASIACRYPT '12**
- *Lattice Signatures and Bimodal Gaussians*
  D., A. Durmus, T. Lepoint et V. Lyubashevsky **CRYPTO '13**

  Implémentation open source : `http://bliss.di.ens.fr`

+ *Ring-LWE in Polynomial Rings.*
  D. et A. Durmus. **PKC' 12.**
+ *Anonymity from Asymmetry : New Constructions for Anonymous HIBE*
  D. **CT-RSA '10**

En marge de la thèse

La cryptographie à base de réseau est **conceptuellement simple**, et peut être abordée **sans pré-requis**.

Cryptris : Jeu vidéo de **médiation scientifique** sur les fondements et l'utilisation de la cryptographie asymétrique.

Mission doctorale en collaboration avec :

- **INRIA :** coordination, médiation, financement
- **digitalcuisine :** scénario, développement
- **Cap Math :** financement

Résumé des contributions

- Nous démontrons par **des attaques pratiques** que les **contre-mesures heuristiques** sont à proscrire
- Nous **analysons** et **optimisons** le facteur limitant de nombreuses constructions (signatures, IBE, HIBE . . .) : l'**échantillonnage gaussien discret**
- Nous démontrons que la cryptographie fondée sur les réseaux peut être rendue **compétitive en pratique**, en particulier pour les **signatures**

Conclusions

Il y a encore 4 ans, les réseaux étaient prometteurs. Ils sont aujourd'hui incontournables ayant ouvert des portes inimaginées.

FHE, Attribute Encryption, Multi Linear Maps, $i\mathcal{O}$

Conclusions

Il y a encore 4 ans, les réseaux étaient prometteurs. Ils sont aujourd'hui incontournables ayant ouvert des portes inimaginées.

FHE, Attribute Encryption, Multi Linear Maps, $i\mathcal{O}$

Après des débuts heuristiques, orientés vers la pratique, le domaine est aujourd'hui très théorique.

Conclusions

Il y a encore 4 ans, les réseaux étaient prometteurs. Ils sont aujourd'hui incontournables ayant ouvert des portes inimaginées.

FHE, Attribute Encryption, Multi Linear Maps, $i\mathcal{O}$

Après des débuts heuristiques, orientés vers la pratique, le domaine est aujourd'hui très théorique.

Un meilleur **équilibre** entre théorie, pratique et cryptanalyse serait bénéfique ;

les **challenges** sont essentiels.

Perspectives

- formaliser le problème du choix de paramètres en pratique et l'**automatiser** ;
- contribuer à la **compréhension** et à l'**amélioration** des techniques de cryptanalyse ;
- **formaliser les analogies** entre les constructions sur réseaux euclidiens et celles sur les courbes elliptiques.

Merci !

Merci à tous pour votre présence et votre attention !

Questions du jury


Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre.
From identification to signatures via the Fiat-Shamir transform : Minimizing assumptions for security and forward-security.
In Lars R. Knudsen, editor,
Advances in Cryptology – EUROCRYPT 2002 Advances in Cryptology – EUROCRYPT , volume 2332 of , pages 418–433. Springer, April / May 2002.


Yuanmi Chen and Phong Q. Nguyen.
BKZ 2.0 : Better lattice security estimates.
In Dong Hoon Lee and Xiaoyun Wang, editors,
Advances in Cryptology – ASIACRYPT 2011 Advances in Cryptology – ASIACRYPT , volume 7073 of , pages 1–20. Springer, December 2011.


Oded Goldreich, Shafi Goldwasser, and Shai Halevi.
Public-key cryptosystems from lattice reduction problems.
In Burton S. Kaliski Jr., editor, , volume 1294 of , pages 112–131. Springer, August 1997.


Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann.
Practical lattice-based cryptography : A signature scheme for embedded systems.

In Emmanuel Prouff and Patrick Schaumont, editors,
Cryptographic Hardware and Embedded Systems – CHES 2012 Cryptographic Hardware a volume 7428 of , pages 530–547. Springer, September 2012.


Nicolas Gama, Phong Q. Nguyen, and Oded Regev.
Lattice enumeration using extreme pruning.
In Henri Gilbert, editor,
Advances in Cryptology – EUROCRYPT 2010 Advances in Cryptology – EUROCRYPT , volume 6110 of , pages 257–278. Springer, May 2010.


Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan.
Trapdoors for hard lattices and new cryptographic constructions.
In Richard E. Ladner and Cynthia Dwork, editors, 40th , pages 197–206. ACM Press, May 2008.


Nick Howgrave-Graham.
A hybrid lattice-reduction and meet-in-the-middle attack against NTRU.
In Alfred Menezes, editor,
Advances in Cryptology – CRYPTO 2007 Advances in Cryptology – CRYPTO , volume 4622 of , pages 150–169. Springer, August 2007.


Jeffrey Hoffstein, Jill Pipher Nick Howgrave-Graham, Joseph H. Silverman, and William Whyte.
NTRUSIGN : Digital signatures using the NTRU lattice.
In Marc Joye, editor,
Topics in Cryptology – CT-RSA 2003 Topics in Cryptology – CT-RSA , volume 2612 of , pages 122–140. Springer, April 2003.


Yupu Hu, Baocang Wang, and Wencai He.
NTRUSign with a new perturbation.
IEEE Transactions on Information Theory, 54(7) :3216–3221, 2008.


Philip N. Klein.
Finding the closest lattice vector when it's unusually close.
In Proc. ACM SODA, pages 937–941, 2000.


Vadim Lyubashevsky.
Fiat-Shamir with aborts : Applications to lattice and factoring-based signatures.
In Mitsuru Matsui, editor,
Advances in Cryptology – ASIACRYPT 2009 Advances in Cryptology – ASIACRYPT ,
volume 5912 of , pages 598–616. Springer, December 2009.


Vadim Lyubashevsky.
Lattice signatures without trapdoors.
In David Pointcheval and Thomas Johansson, editors,
Advances in Cryptology – EUROCRYPT 2012 Advances in Cryptology – EUROCRYPT ,
volume 7237 of , pages 738–755. Springer, April 2012.


Daniele Micciancio and Petros Mol.
Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions.
In Phillip Rogaway, editor,
Advances in Cryptology – CRYPTO 2011 Advances in Cryptology – CRYPTO , volume 6841 of , pages 465–484. Springer, August 2011.


Daniele Micciancio and Chris Peikert.
Trapdoors for lattices : Simpler, tighter, faster, smaller.
In David Pointcheval and Thomas Johansson, editors,
Advances in Cryptology – EUROCRYPT 2012 Advances in Cryptology – EUROCRYPT , volume 7237 of , pages 700–718. Springer, April 2012.


T. Malkin, C. Peikert, R. A. Servedio, and A. Wan.
Learning an overcomplete basis : Analysis of lattice-based signatures with perturbations.
2009 manuscript cited in [Pei10], available as [Wan10, Chapter 6].


Phong Q. Nguyen and Oded Regev.
Learning a parallelepiped : Cryptanalysis of GGH and NTRU signatures.
In Serge Vaudenay, editor,
Advances in Cryptology – EUROCRYPT 2006 Advances in Cryptology – EUROCRYPT , volume 4004 of , pages 271–288. Springer, May / June 2006.


Chris Peikert.
An efficient and parallel gaussian sampler for lattices.
In Tal Rabin, editor,
Advances in Cryptology – CRYPTO 2010 Advances in Cryptology – CRYPTO , volume 6223 of , pages 80–97. Springer, August 2010.


A. Wan.
Learning, cryptography, and the average case.
PhD thesis, Columbia University, 2010.
Available at http://itcs.tsinghua.edu.cn/ atw12/.