Cryptographie post-quantique basée de les réseaux

Léo Gagnon – Juillet 2019

Pourquoi?

$$329317 \times 988061 = 325385284337$$

$$17 = 5^{13} \; mod \; 19$$

Pourquoi?

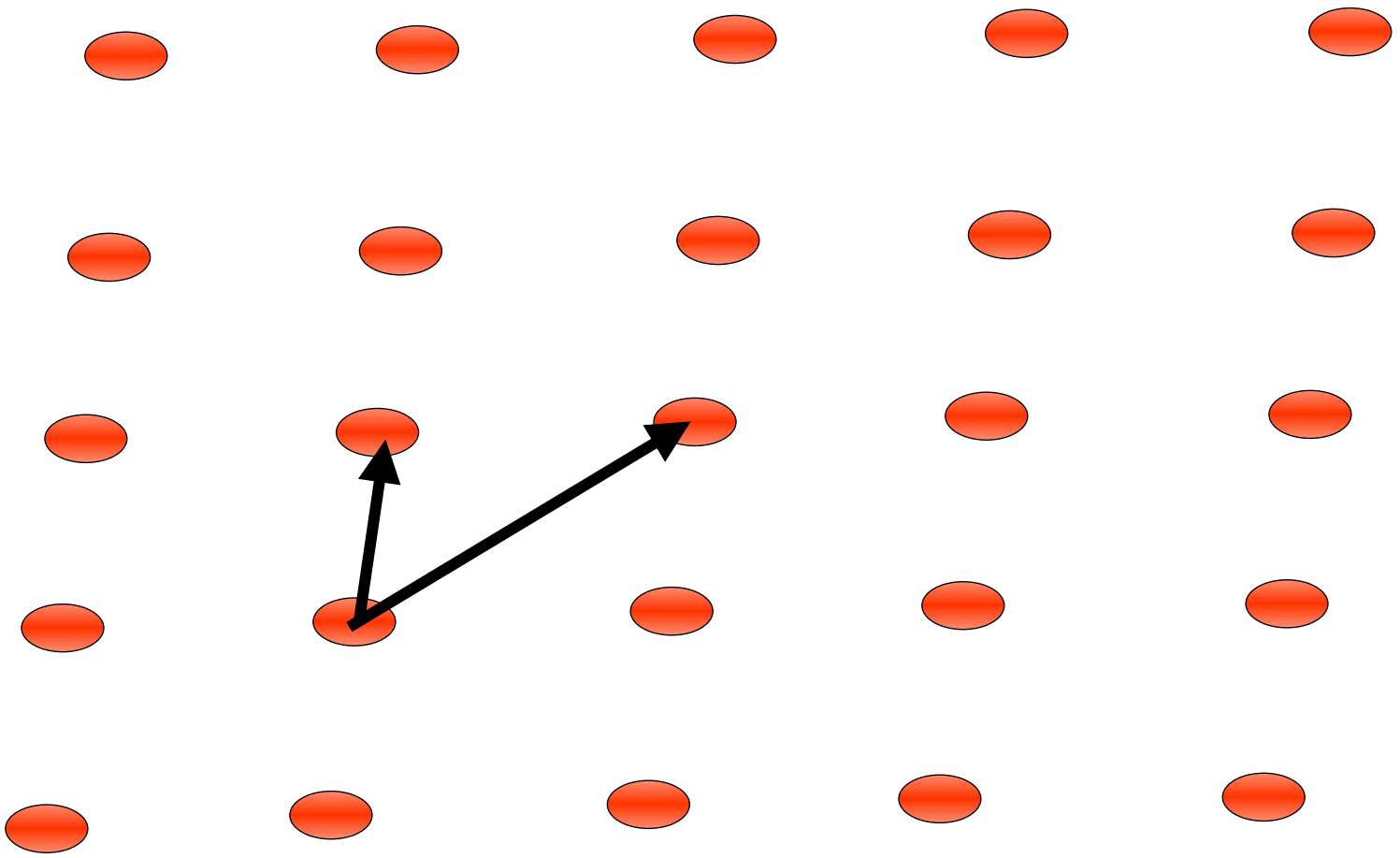
Réseaux euclidiens

Un réseau Λ de dimension n est un sous-groupe discret additif de $\mathbb{R}^n$. Autrement dit,

1. $\mathbf{0} \in \Lambda$ et $-x, x+y \in \Lambda \; \forall x, y \in \Lambda$
2. Chaque $x \in \Lambda$ possède un voisinage dans $\mathbb{R}^n$ dans lequel il est le seul élément de Λ

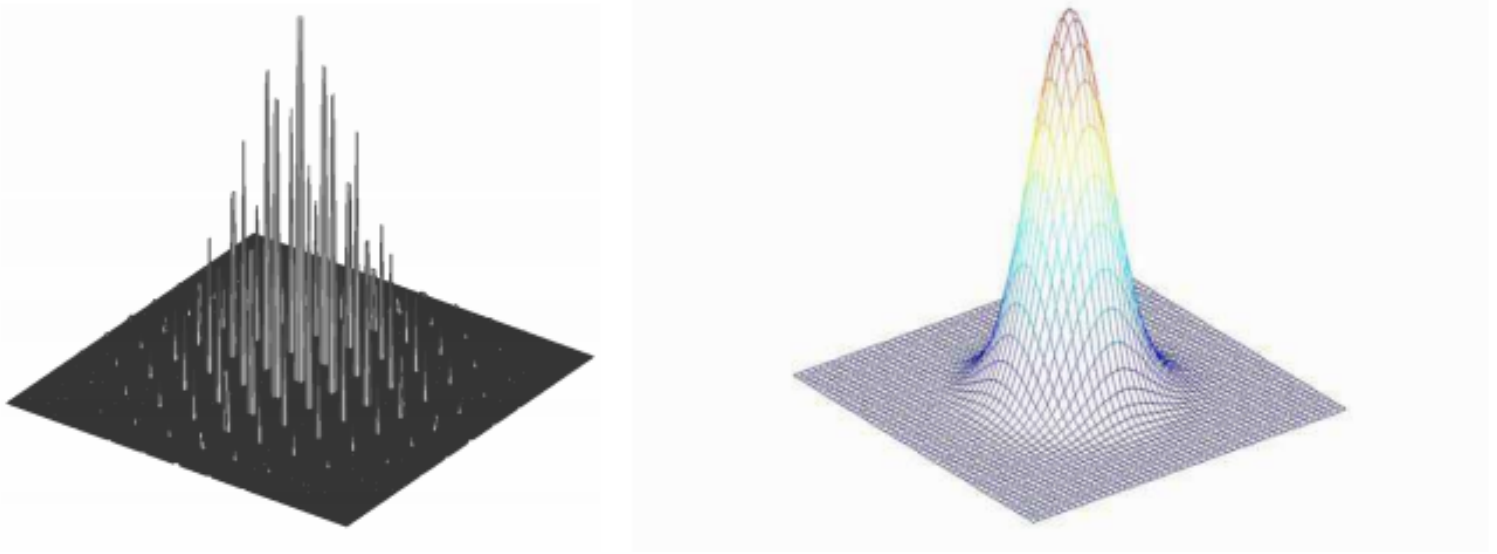
L'exemple typique de réseau est $\mathbb{Z}^n$.
Pour spécifier explicitement un réseau de dimension n, il suffit de donner n vecteurs indépendants $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ tel que $\Lambda = \{\sum_{i=1}^{n} z_i \mathbf{b}_i : z_i \in \mathbb{Z}\}$ (on appelle ces vecteurs la base de Λ).

Réseaux euclidiens



Réseaux euclidiens

On défini la loi normale discrète sur Λ comme la loi normale de moyenne 0 et variance r sur $\mathbb{R}^n$ discrétisé aux points de Λ.

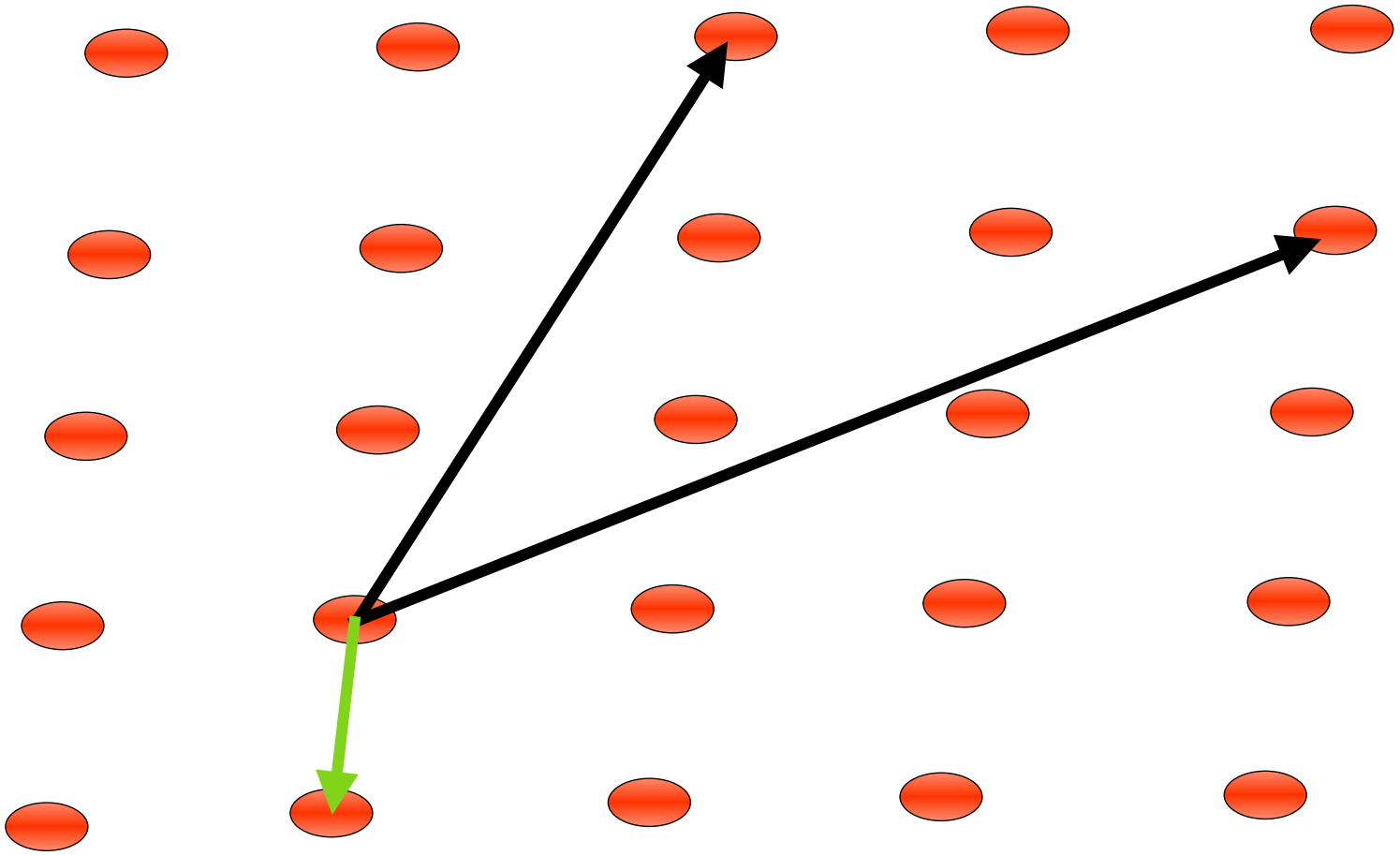


Problèmes calculatoires classiques sur les réseaux

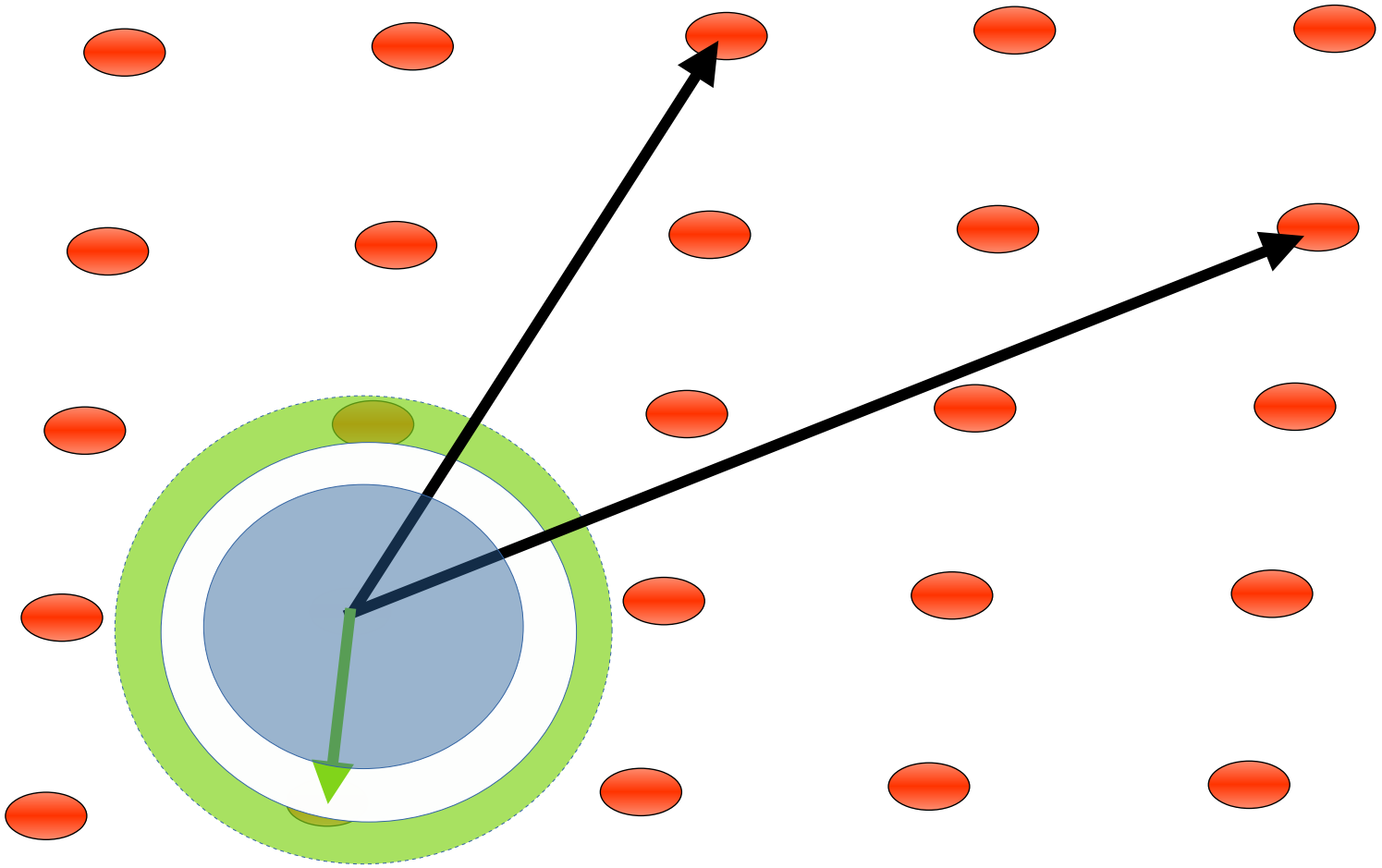
Shortest vector problem : (Gap)SVP

Soit un réseau Λ, SVP demande de trouver le plus petit vecteur $\mathbf{v}$ de Λ tant dis que GapSVP demande de distinguer entre les cas $\mathbf{v} \leq d$ et $\mathbf{v} > d\gamma$. Les deux problèmes sont équivalents. Aussi, il est à noter que GapSVP est facile pour des valeurs exponentielles de γ et NP-difficile pour des petites valeurs de γ, la cryptographie s'intéresse à des $\gamma = poly(n)$. Ce problème est conjecturé difficile (jusqu'à maintenant) même pour les ordinateurs quantique et les meilleurs algorithmes connus fonctionnent en temps $2^{O(n)}$ pour des facteurs d'approximation polynomiaux.

Shortest vector problem : (Gap)SVP



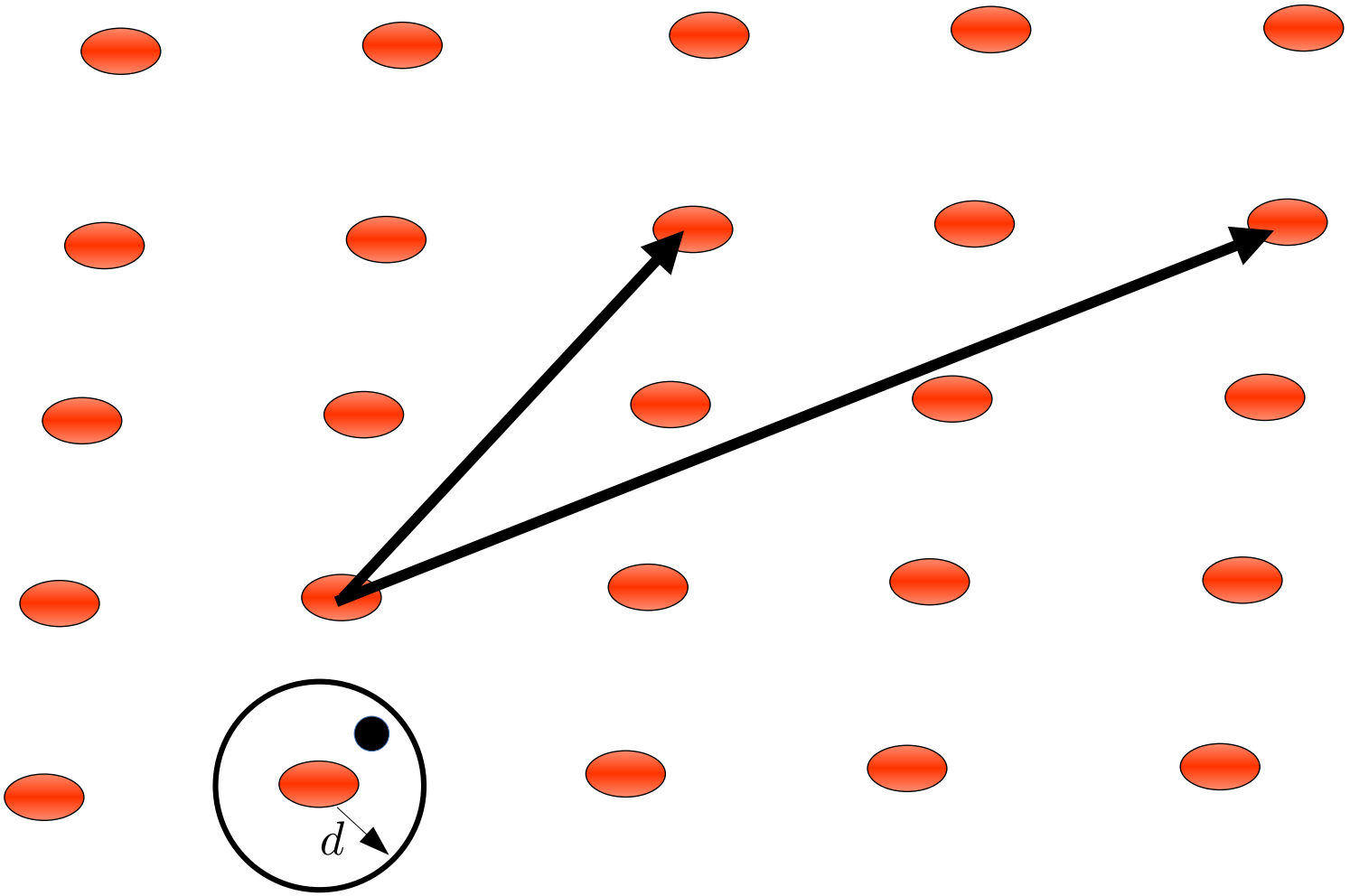
Shortest vector problem : (Gap)SVP



Bounded distance decoding : BDD

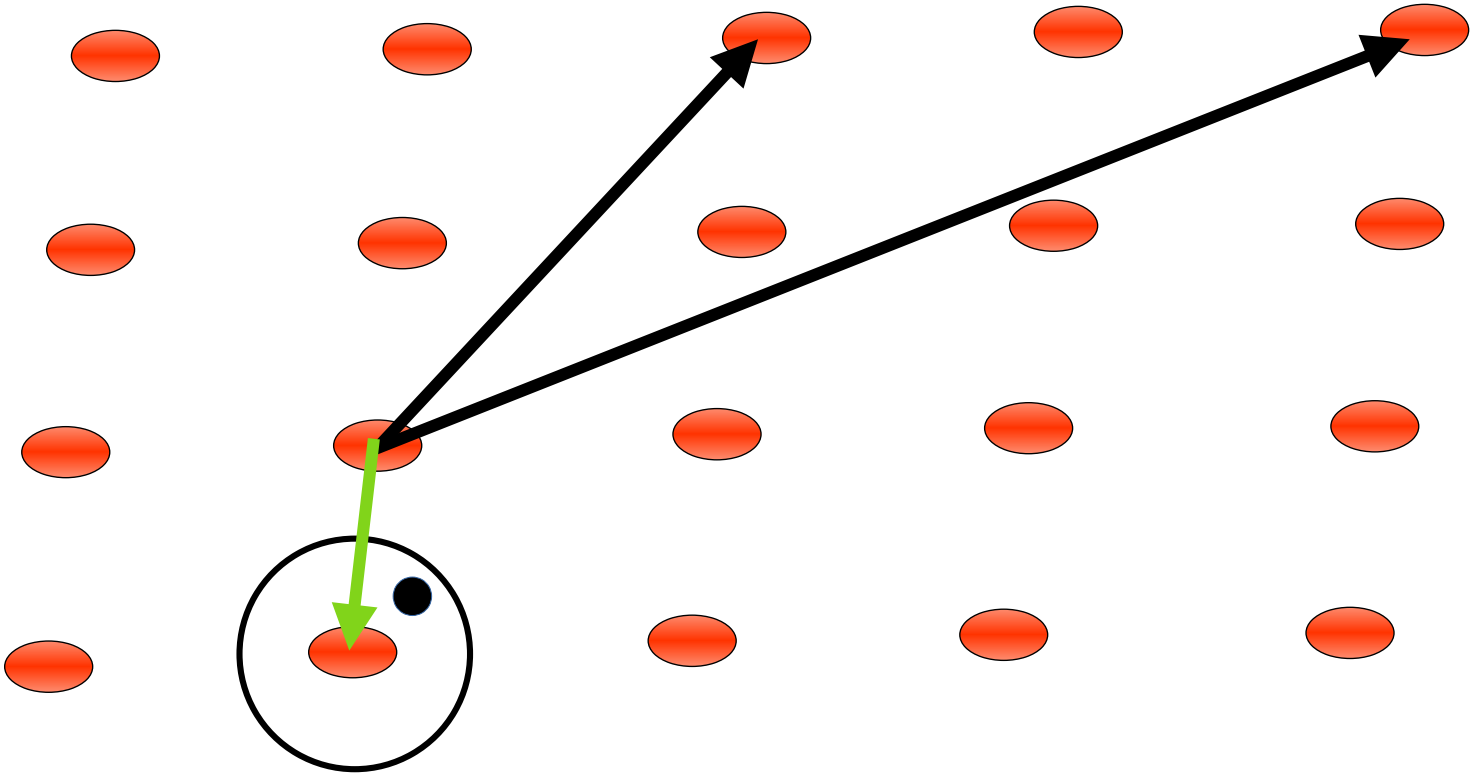
Soit un réseaux Λ et un point $\mathbf{x}$ à une distance maximale $d > 0$ de Λ (le plus court vecteur reliant $\mathbf{x}$ à un point $\mathbf{y} \in \Lambda$ a une norme inférieure à d). L'objectif est de trouver le point $\mathbf{y} \in \Lambda$ le plus près (distance euclidienne) de $\mathbf{x}$. Remarquons la réponse est unique si et seulement si $d < \lambda_1(\Lambda)/2$. Le meilleur algorithme connu pour BDD fonctionne en temps exponentiel.

Bounded distance decoding : BDD

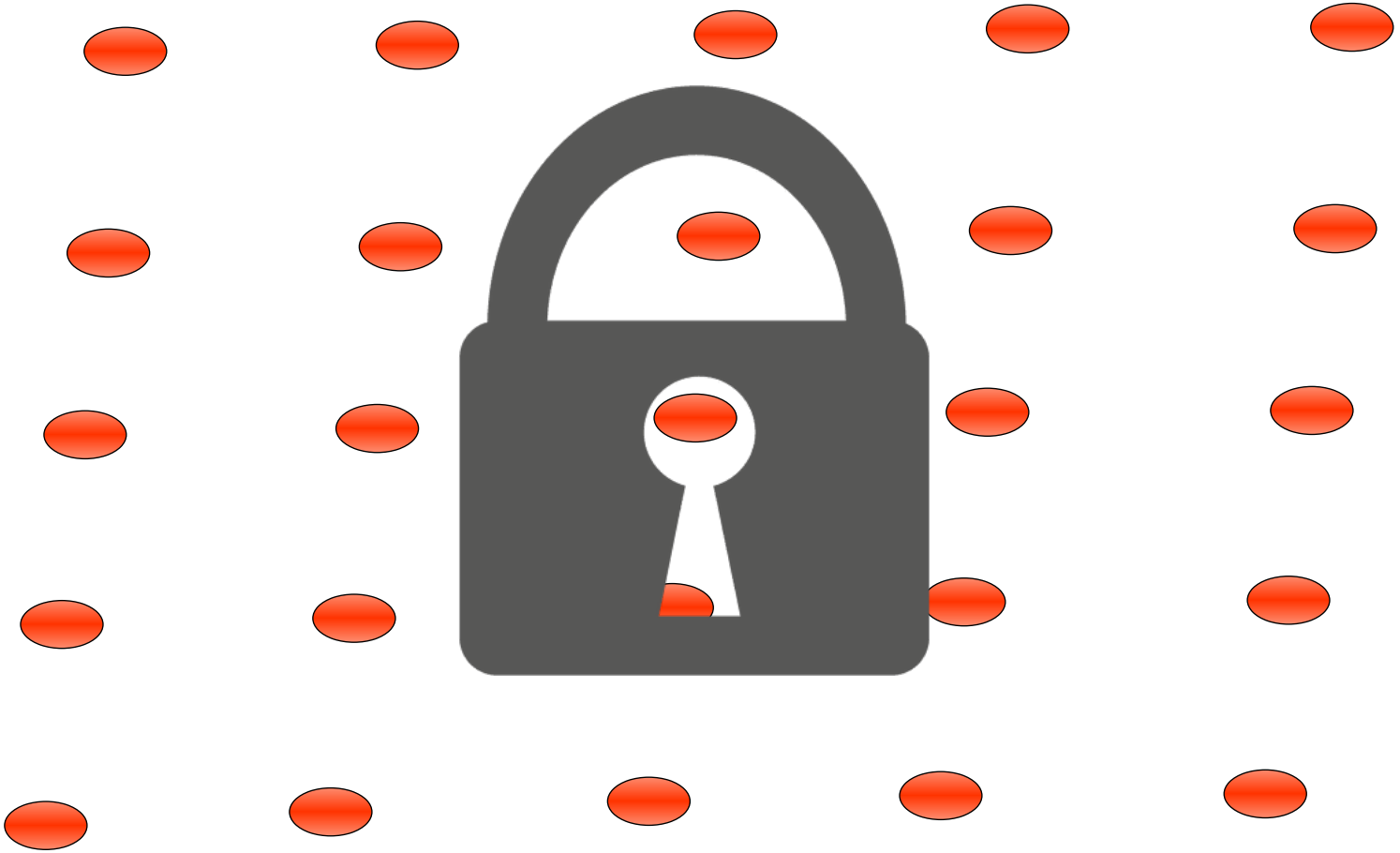


Bounded distance decoding : BDD

On peut montrer que si on était capable de résoudre BDD pour de petites valeurs de d, alors on serait capable de résoudre GapSVP pour $\gamma = poly(n)$. Ainsi GapSVP $\leq BDD$.



Problèmes calculatoires utiles à la cryptographie



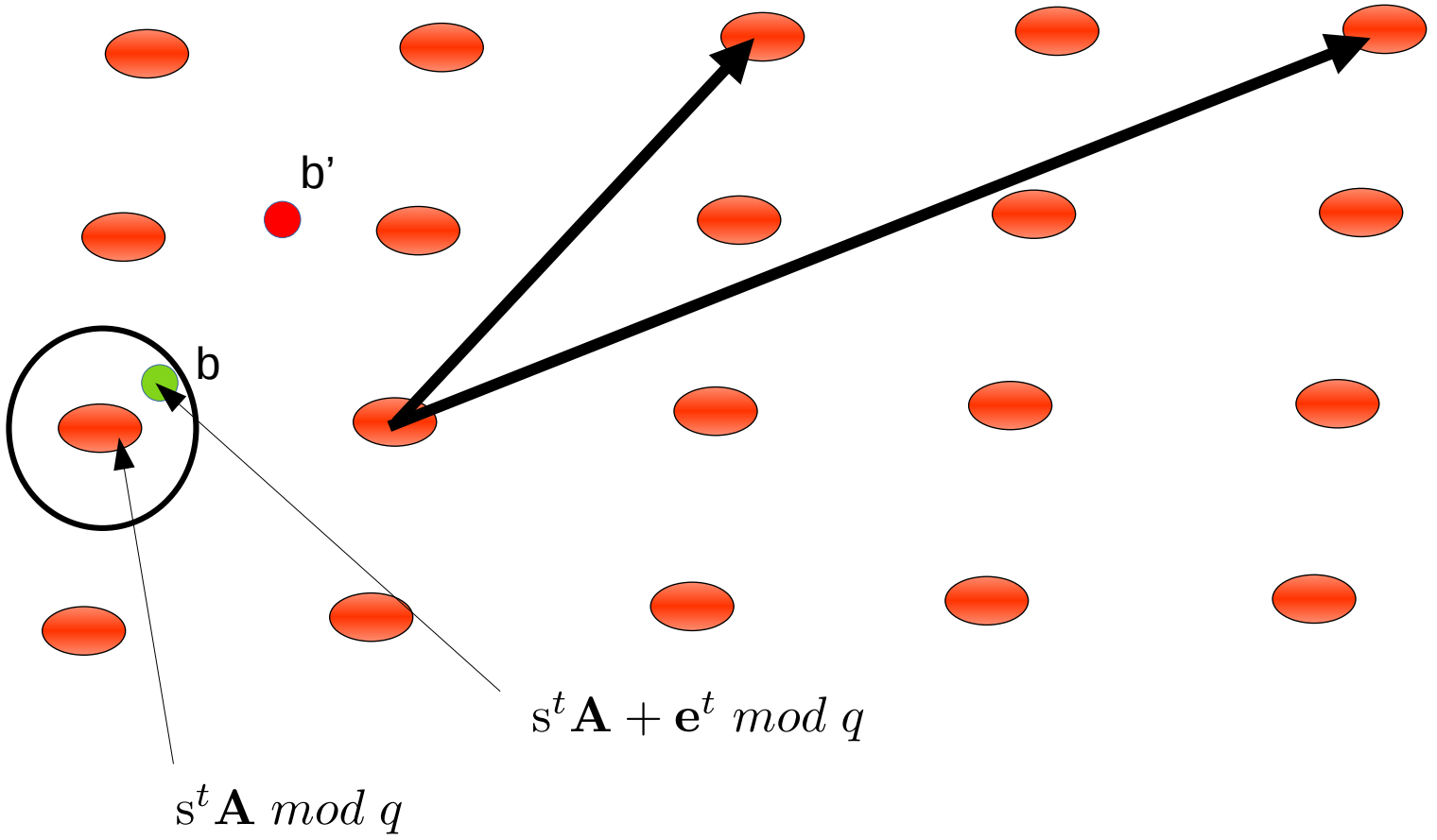
Learning with errors : LWE

Soient $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \chi^m$ (une distribtuion normale $(0, \alpha q)$) et $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Étant donné $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \ mod\ q$ trouver $\mathbf{s}$. Une version décisionnelle (souvent utilisée pour la cryptographie) demande de distringuer $(\mathbf{A}, \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \ mod\ q)$ de $(\mathbf{A}, \mathbf{b}^t)$ uniforme.

$$\mathbf{A} = \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_m \\ | & & | \end{pmatrix}, \ \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$$

Learning with errors : LWE

Il est facile de voir que résoudre LWE avec la matrice $\mathbf{A}$ équivaut à résoudre BDD sur $\Lambda = \{\mathbf{z} = \mathbf{s}^t\mathbf{A} \; mod \; q \mid \mathbf{s} \in \mathbb{Z}_q^n\}$. Donc GapSVP $\leq BDD \leq LWE$.



Learning with errors : LWE

À propos de l'erreur, elle provient d'une loi normale de moyenne 0 et de variance αq où $\alpha \in (0, 1)$ est choisi en fonction de la sécurité désirée. La réduction montre que avec un oracle pour LWE_α, on peut résoudre GapSVP avec un facteur d'approximation de $O(n/\alpha)$. On choisi donc en général $\alpha \geq 1/poly(n)$.

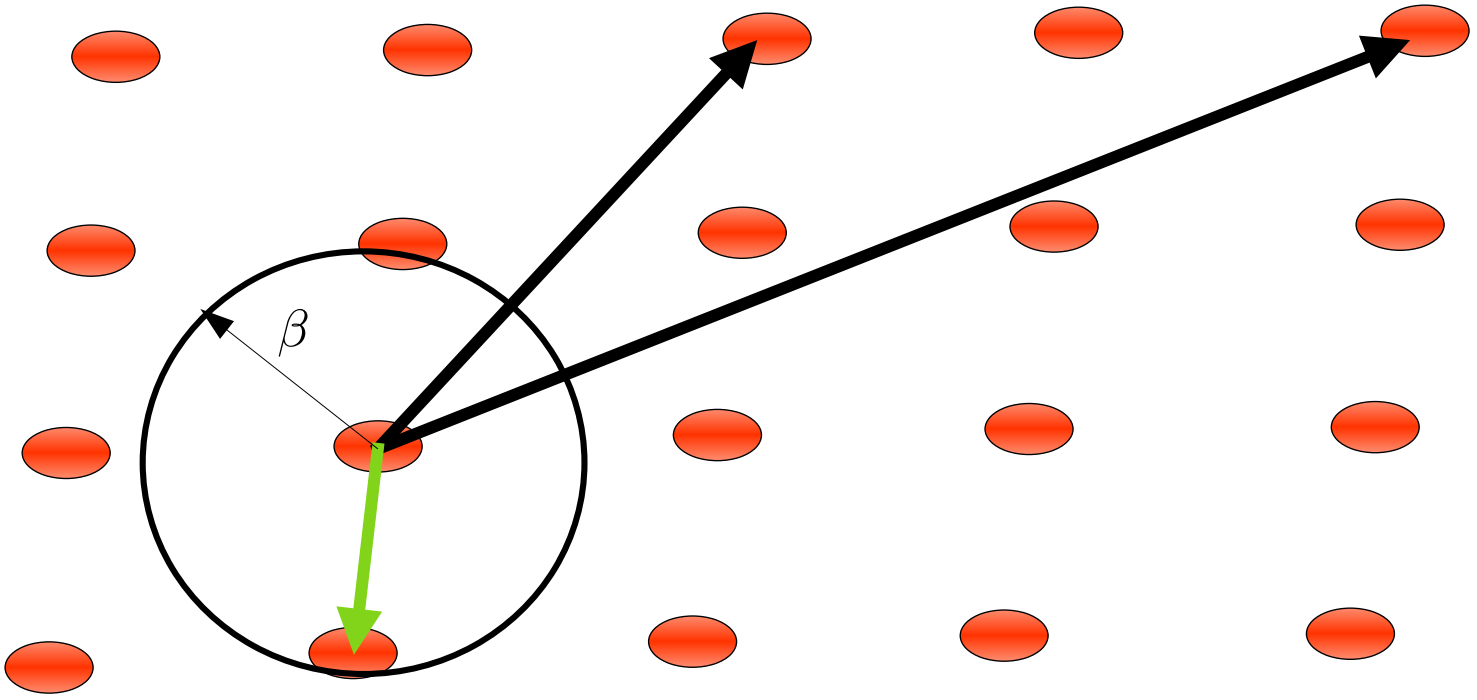
Shorter integer solution : SIS

Soit $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ une matrice $n \times m$ constituée de m vecteurs aléatoires choisis uniformément dans . L'objectif est de trouver $\mathbf{z} \in \mathbb{Z}^m$ non-nul tel que $||\mathbf{z}|| \leq \beta$ et $f_{\mathbf{A}}(\mathbf{z}) := \mathbf{A}\mathbf{z}\ mod\ q = \mathbf{0}$.

$$\underbrace{\begin{pmatrix} \cdots & \mathbf{A} & \cdots \end{pmatrix}}_{m} \begin{pmatrix} \\ \mathbf{z} \\ \\ \end{pmatrix} = \mathbf{0} \in \mathbb{Z}_q^n$$

Shorter integer solution : SIS

L'ensemble $\Lambda = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = 0 \ mod \ q\}$ (le noyeau de $\mathbf{A}$) forme un réseau, donc résoudre SIS avec la matrice $\mathbf{A}$ revient à trouver un petit vecteur de Λ (SVP/GapSVP). Ainsi, GapSVP $\leq$ SIS avec un facteur d'approximation $\beta \cdot poly(n)$.



Applications cryptographiques !

Applications cryptographiques !

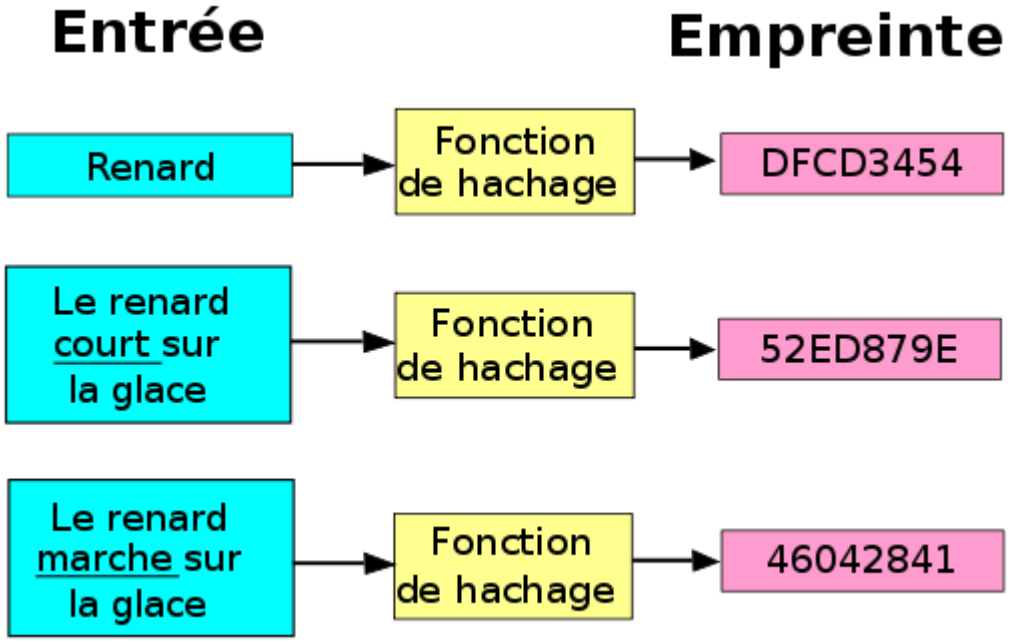
SIS :

- Trouver $\mathbf{z} \neq 0$ court tel que $\mathbf{Az}\ mod\ q = \mathbf{0}$
- Problème calculatoire
- Applications : fonction à sens unique, fonction de hachage résistante aux collisions et autres "outils" cryptographiques.

LWE :

- Distinguer $(\mathbf{A}, \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t\ mod\ q)$ de $(\mathbf{A}, \mathbf{b}^t)$ uniforme.
- Problème de décision
- Applications : encryption à clée privée, encryption à clée publique et beaucoup plus.

Fonction de hachage résistante au collisions



Résistance au collisions : étant donné une fonction de hachage $f(x)$, il est impossible pour un adversaire efficace de trouver x et x' tels que $f(x) = f(x')$ sauf avec probabitlié négligeable.

Fonction de hachage résistante aux collisions

Soit $h_{\mathbf{A}}$: $\{0, \ldots, \beta\}^m \rightarrow$ tel que:

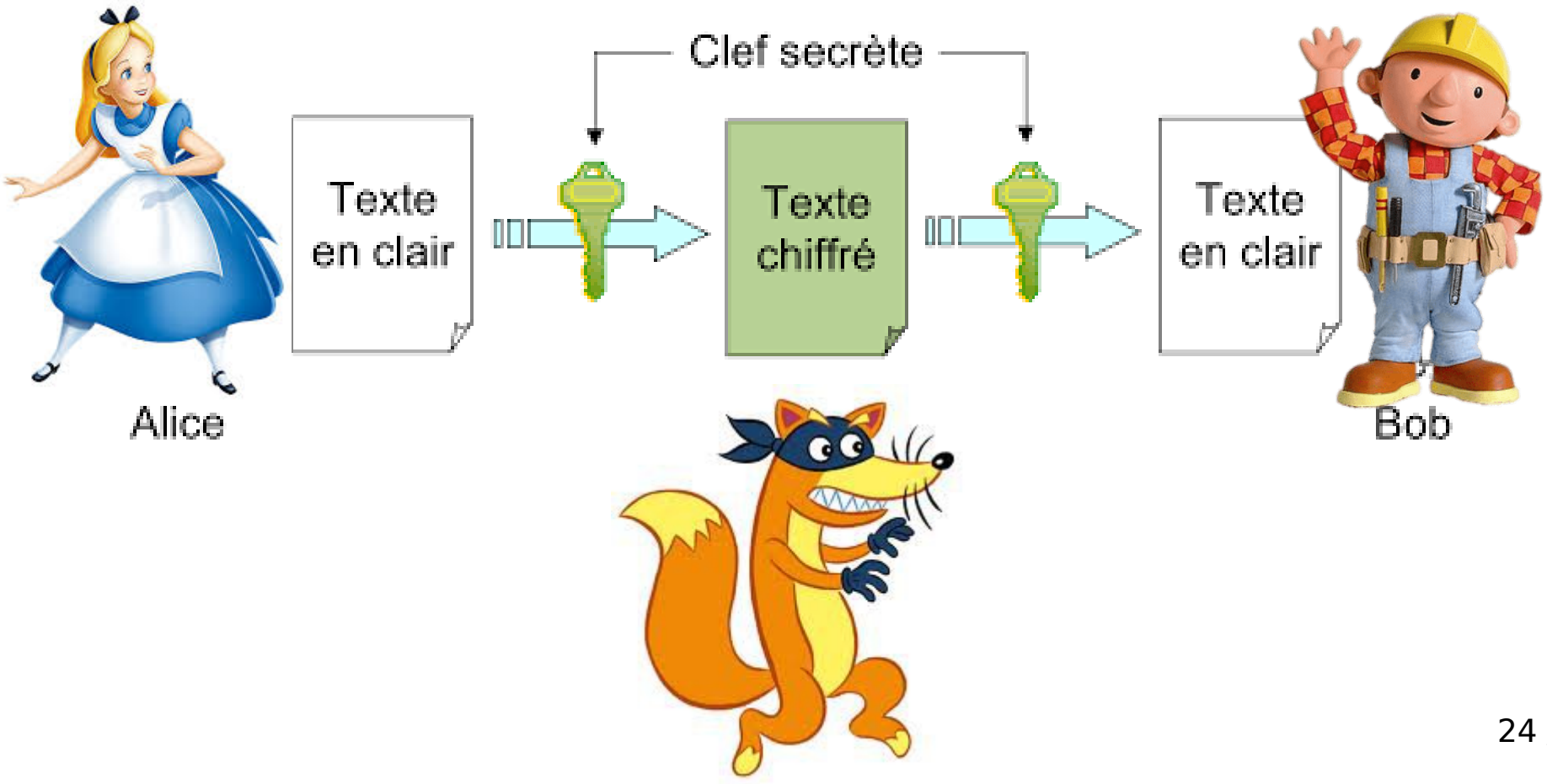
$$h_{\mathbf{A}}(\mathbf{z}) = \mathbf{A}\mathbf{z} \; mod \; q$$

où $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Une collision dans $h_{\mathbf{A}}$ donne $\mathbf{z}, \mathbf{z}' \in \{0, \ldots, \beta\}^m$ tels que $\mathbf{A}(\mathbf{z} - \mathbf{z}') = 0 \; mod \; q$. On a donc que $\mathbf{z} - \mathbf{z}' \in \{-\beta, \ldots, \beta\}^m$ est une solution pour SIS_β.

Avec une fonction de hachage pour un input de taille fixe on peut créer une fonction de hachage pour un input de taille variable

Cryptographie à clée privée

$$(Gen, Enc, Dec)$$



Cryptographie à clée privée

L'espace de message est $\mathcal{M} := \{0, 1\}$.

- Choissir la clée privée
$$k := \mathbf{s} \in \mathbb{Z}_q^n$$
aléatoirement de façon uniforme.

- Générer le cryptogramme
$$c := (\mathbf{a}, b) := (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e + m\lfloor q/2 \rceil \ mod \ q)$$
où $\mathbf{a} \in \mathbb{Z}_q^n$ et $e \in \chi$ (et $|e| < q/4$) son choisis aléatoirement de façon uniforme.

- Retourner $m = 0$ si
$$|b - \langle \mathbf{a}, \mathbf{s} \rangle \ mod \ q| = |e + m\lfloor q/2 \rceil \ mod \ q| < q/4$$
et 1 sinon.

Inefficacité des constructions :(

Les constructions cryptographiques de base basées sur les problèmes de réseaux sont très peu efficaces pour les raisons suivantes :

1. Grandes clées et beaucoup d'aléat. (ex. cryptogramme $O(n)$ fois plus long que le message pour le système à clée privée et matrice $n \times m$ pour la fonction de hachage).

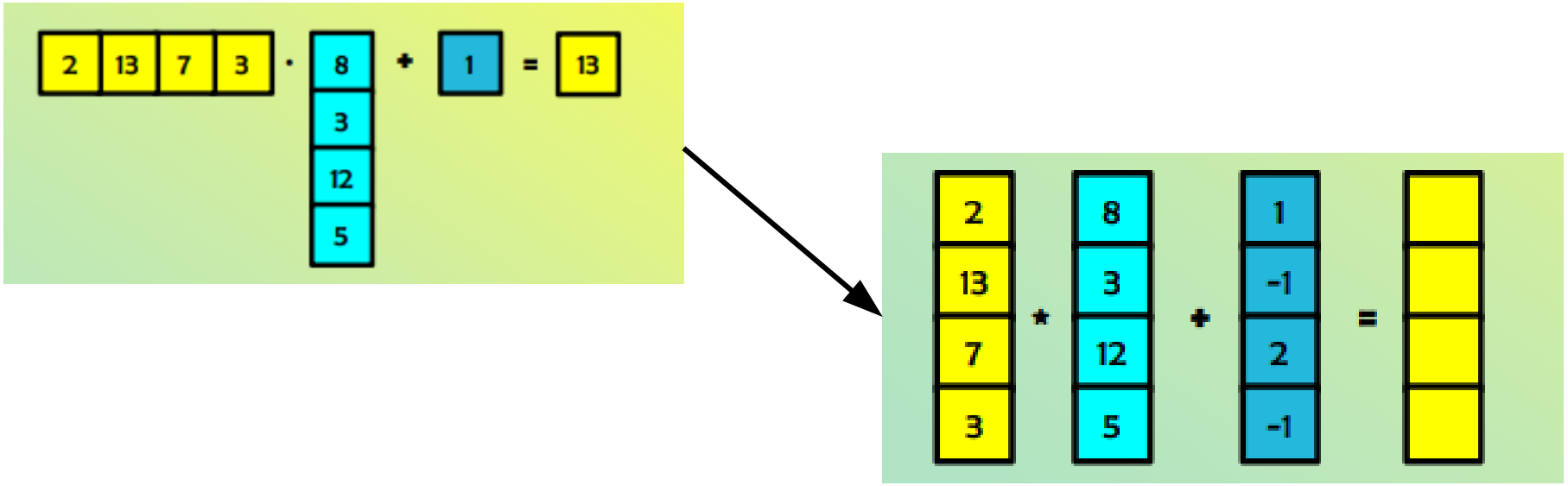
2. Les vecteurs et matrices de grande taille utilisés dans LWE et SIS prennent beaucoup de mémoire.

3. Les multiplications de ces vecteurs/matrices sont peu efficaces.

Il faut donc revoir les constructions si on veux pouvoir les appliquer dans la "vraie vie".

Solution : Redéfinition sur des anneaux

Pour régler ce problème, nous utiliserons l'anneau $\mathbb{Z}_q[x]/\langle x^n+1\rangle$: l'ensemble polynômes résidus modulo (x^n+1) et q (pour les coefficients).



La multiplication dans $\mathbb{Z}_q[x]/\langle x^n+1\rangle$ peut être interpreté comme un produit de convolution discret et celui-ci peut être calculé très efficacement $O(n \log n)$ avec la transformation de Fourirer rapide (vs $O(n^2)$ pour $\mathbb{Z}_q^n$).

Solution : Redéfinition sur des anneaux

En utilisant $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ au lieu de $\mathbb{Z}_q^n$ on obtient donc les résultats suivants.

1. Beaucoup moins d'aléat est nécéssaire.
2. On a besoins de moins gros objets mathématiques pour générer la même quantité de nombres pseudo-aléatoires.
3. Opération dans $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ très efficace.
4. Et surtout LWE et SIS restent difficile même définis sur R_q (ring-LWE et ring-SIS).

Youpi!

Redéfinition des problèmes

ring-SIS

Étant donné $a_1, \ldots, a_l \in R_q$ tous choisis aléatoirement de façon uniforme, le but est de trouver $e_1, \ldots, e_l \in R$ non tous-nuls tels que $a_1 e_1 + \ldots + a_l e_l = 0 \; mod \; qR$ et les e_i sont "petits".

ring-LWE

Étant donné des échantillions $(a, b = a \cdot s + e) \in R_q \times R_q$, où $s \in R_q$ est le secret, $a \in R_q$ est choisi uniformément et e est un terme d'erreur choisi selon χ (une loi normale), le but est de retrouver s. La version décisionnelle demande de distinguer $(a, b = a \cdot s + e)$ de (a, b) tiré d'une distribution uniforme sur $R_q \times R_q$.

Fonction de hachage efficace

$$h_{a_1,\ldots,a_l}(e_1,\ldots,e_l) = a_1e_1 + \ldots + a_le_l \; mod \; qR \text{ où } l = m/n.$$

Clée de taille m au lieu de $m \times n$ et plus efficace !

SWIFFT: A Modest Proposal for FFT Hashing*

Vadim Lyubashevsky[1], Daniele Micciancio[1], Chris Peikert[2,**], and Alon Rosen[3]

[1] University of California at San Diego
[2] SRI International
[3] IDC Herzliya

Encryption à clée privée efficace

L'espace de message est $\mathcal{M} := R_2$ (éléments de R avec coefficients 0,1).

- Choissir la clée privée
$$k := s \in R_q$$
aléatoirement de façon uniforme.

- Générer le cryptogramme
$$c := (a, b) := (a, a \cdot s + e + m\lfloor q/2 \rceil \ mod \ qR)$$
où $a \in R_q$ et $e \in \chi$ ($|e| < q/4$) son choisis aléatoirement de façon uniforme.

- Calculer
$$\hat{m} = b - a \cdot s \ mod \ qR = m\lfloor q/2 \rceil + e$$
Arrondir chaque coefficient de $\hat{m}$ à 0 ou $q/2$. Interpréter 0 comme 0 et $q/2$ comme 1.

Encryption à clée privée efficace

Encryption de $O(n)$ bits avec $O(n)$ bits d'aléat et plus efficace!

Autres applications

- Encryption à clée publique
- Fonction pseudo-aléatoire
- Encryption complètement homomorphe
- Schéma de signature numérique
- Encryption basée sur l'identitée
- Échange de clées
- ...

Applications concrètes

SPRING: Fast Pseudorandom Functions from Rounded Ring Products

Abhishek Banerjee[1⋆], Hai Brenner[2⋆⋆], Gaëtan Leurent[3], Chris Peikert[1⋆⋆⋆], and Alon Rosen[2†]

[1] Georgia Institute of Technology
[2] IDC Herzliya
[3] INRIA Team SECRET

FALCON: Fast-Fourier Lattice-based Compact Signatures over NTRU

Specifications v1.0

Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang

CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM

Joppe Bos*, Léo Ducas†, Eike Kiltz‡, Tancrède Lepoint§, Vadim Lyubashevsky¶, John M. Schanck‖, Peter Schwabe**, Gregor Seiler††, Damien Stehlé‡‡,

SWIFFT: A Modest Proposal for FFT Hashing⋆

Vadim Lyubashevsky[1], Daniele Micciancio[1], Chris Peikert[2,⋆⋆], and Alon Rosen[3]

[1] University of California at San Diego
[2] SRI International
[3] IDC Herzliya

● ● ●

Competition du NIST : 2e tour

Cryptographie à clée publique : 18 propositions

1. 7 basées sur la théorie des codes
2. 9 basées sur les réseaux
3. 1 basée sur des courbes elliptiques super-singulière (?)

Systèmes de signature numérique : 11 propositions

1. 3 basées sur les réseaux
2. 4 basées sur des polynômes multivariée (?)
3. 2 basées sur (?)

Conclusion

Encore BEACOUP de questions ouvertes (meilleures réductions, généralisation à plus d'anneaux, attaques).

Lectures intéressantes :

- Chris Peikert : A decade of lattice cryptography
- Chris Peikert : To cyclicity and beyond (https://bit.ly/2Jm3jye)
- Page de la compétition du NIST (https://csrc.nist.gov/Projects/Post-Quantum-Cryptography)

Questions?