Cryptographie et réseaux euclidiens

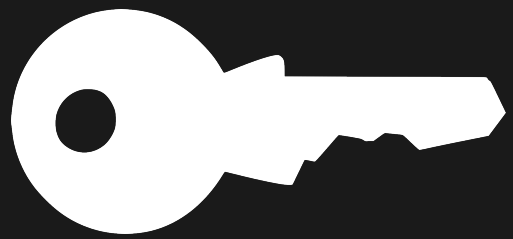
Alice Pellet--Mary

CNRS et Université de Bordeaux

Diffusion classes prépa

Bordeaux

Cryptographie



Cryptographie

(Cryptologie =) Cryptographie = science des secrets

Exemples : chiffrement, signature électronique, vote électronique ...

Cryptographie

(Cryptologie =) Cryptographie = science des secrets

Exemples : chiffrement, signature électronique, vote électronique ...

Applications :

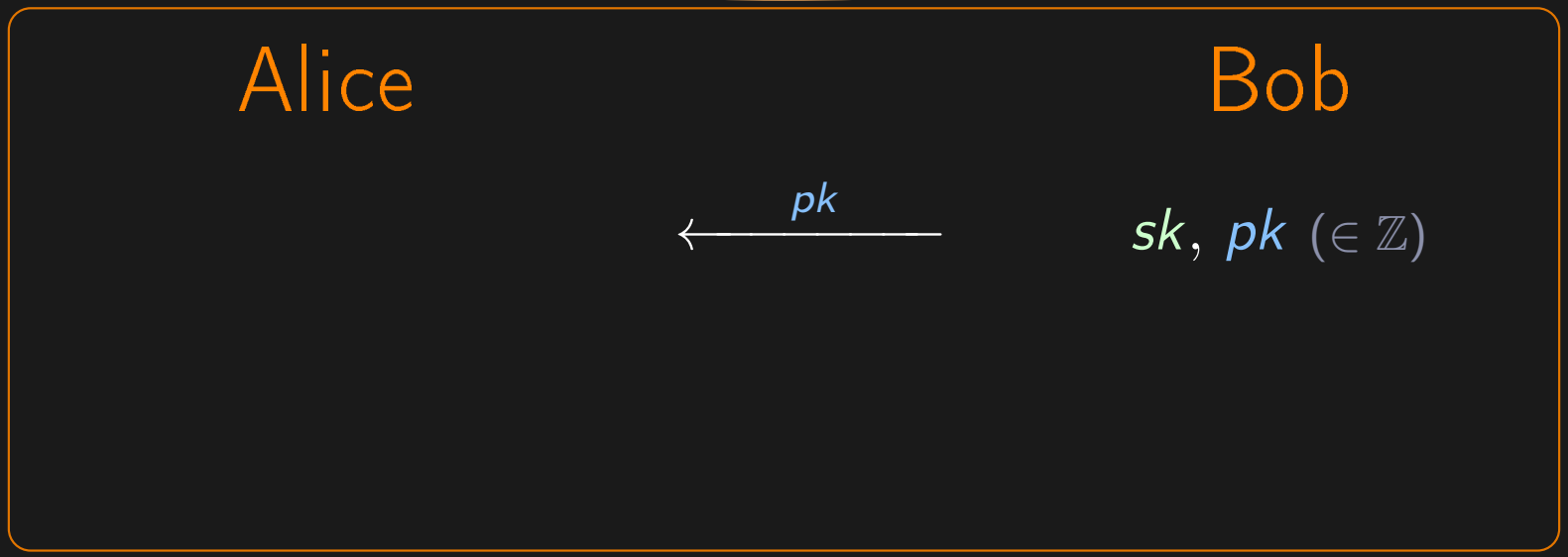
Cryptographie

(Cryptologie =) Cryptographie = science des secrets

Exemples : chiffrement, signature électronique, vote électronique ...

Applications : https, Whatsapp, vote à l'étranger, cryptomonnaies, militaires ...

Chiffrement asymétrique [DH76, RSA78]



[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

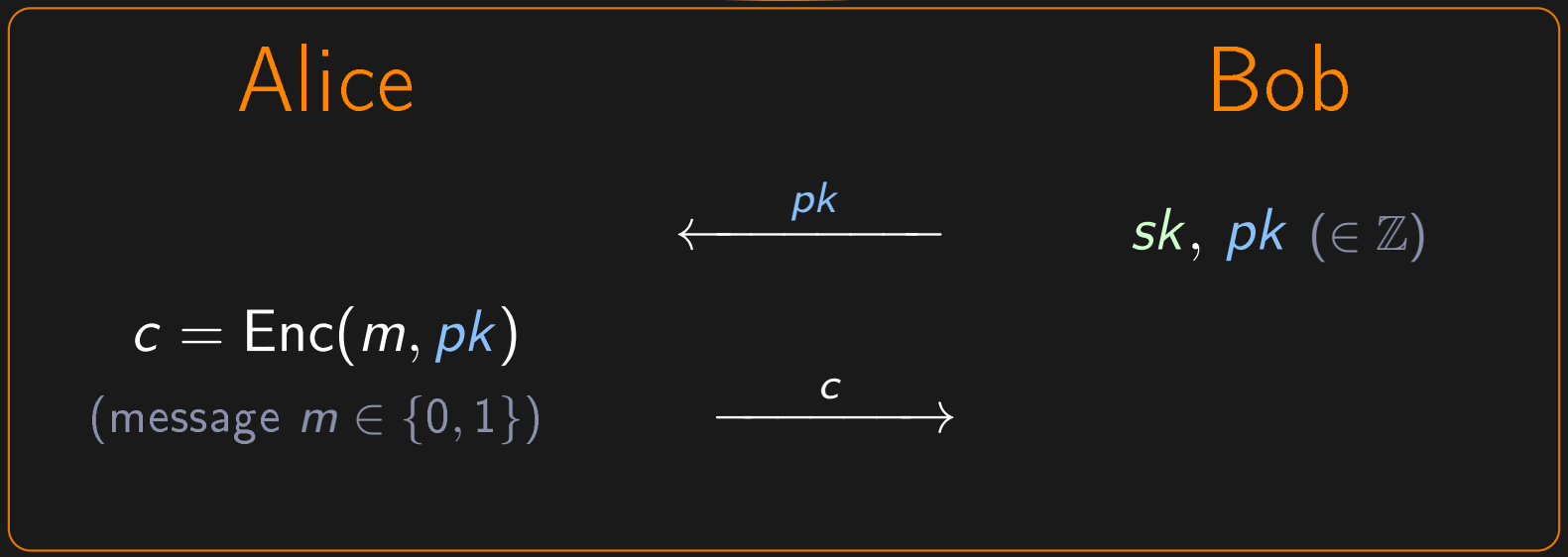
Chiffrement asymétrique [DH76, RSA78]

Alice		Bob
	$\xleftarrow{pk}$	sk, pk $(\in \mathbb{Z})$
$c = \mathsf{Enc}(m, pk)$		
(message $m \in \{0,1\}$)		

[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

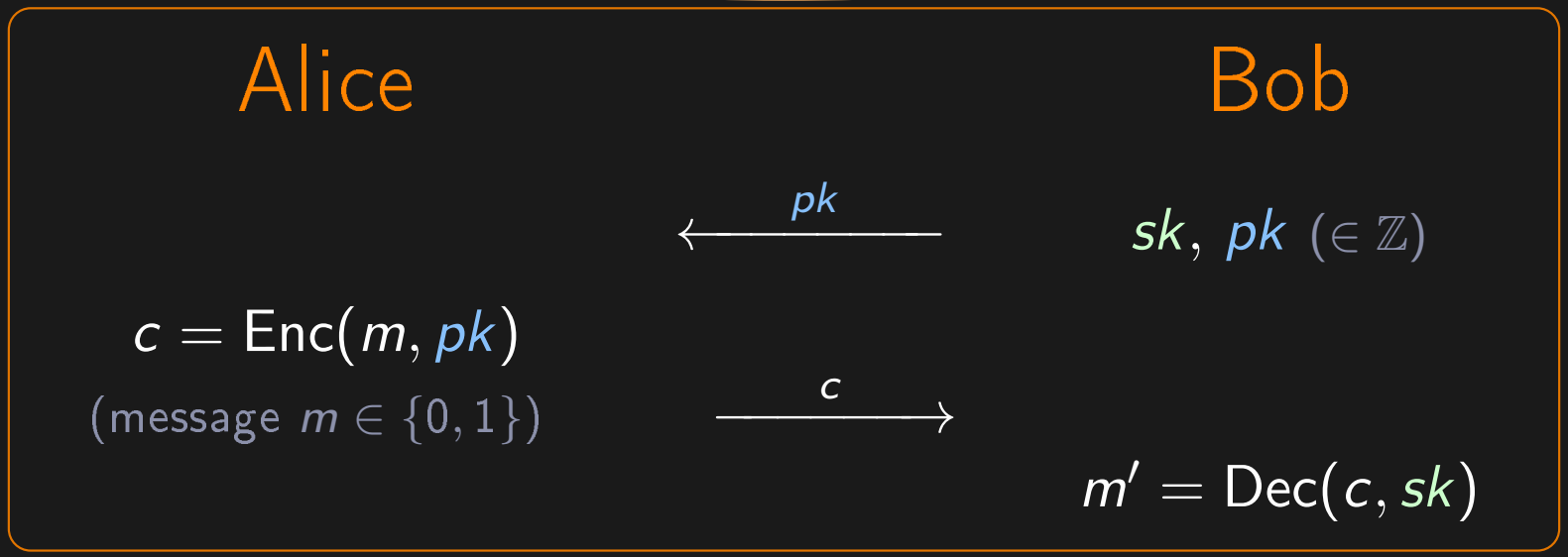
Chiffrement asymétrique [DH76, RSA78]



[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

Chiffrement asymétrique [DH76, RSA78]



[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

Chiffrement asymétrique [DH76, RSA78]

Alice		Bob
	$\xleftarrow{pk}$	$sk, pk \ (\in \mathbb{Z})$
$c = \mathsf{Enc}(m, pk)$ (message $m \in \{0,1\}$)	$\xrightarrow{c}$	$m' = \mathsf{Dec}(c, sk)$

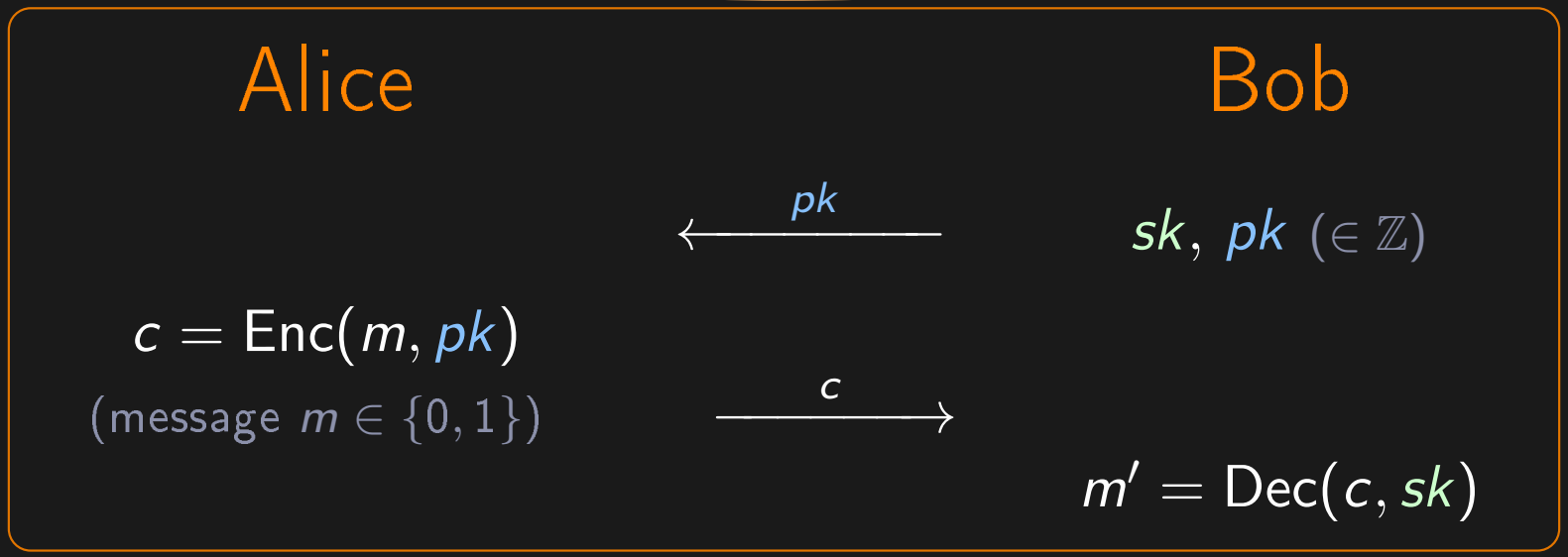
Correction:

$$\forall m \in \{0,1\}, \quad \mathsf{Dec}\Big(\mathsf{Enc}(m, pk), sk\Big) = m$$

[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

Chiffrement asymétrique [DH76, RSA78]



Correction:

$$\forall m \in \{0,1\}, \quad \mathrm{Dec}\Big(\mathrm{Enc}(m, pk), sk\Big) = m$$

Sécurité:

Pour tout algorithme $\mathcal{A}$ qui termine en temps polynomial,

$$\Pr_{m \leftarrow \mathcal{U}(\{0,1\})}\Big(\mathcal{A}\big(pk, \mathrm{Enc}(m, pk)\big) = m\Big) = \frac{1}{2} \pm 2^{-128}$$

[DH76] Diffie, Hellman. New Directions in Cryptography.

[RSA78] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

Une sécurité prouvée ?

On ne connaît pas de construction de chiffrement
dont on peut prouver qu'elle est sûre

Une sécurité prouvée ?

On ne connaît pas de construction de chiffrement
dont on peut prouver qu'elle est sûre

Solution : on fait reposer la sécurité sur la difficulté supposée de certains problèmes algorithmiques.

$\Rightarrow$ ce sont les "axiomes" de la cryptographie

Une sécurité prouvée ?

On ne connaît pas de construction de chiffrement
dont on peut prouver qu'elle est sûre

Solution : on fait reposer la sécurité sur la difficulté supposée de certains problèmes algorithmiques.

$\Rightarrow$ ce sont les "axiomes" de la cryptographie

Exemples : factorisation, logarithme discret, ...
décodage dans un réseau euclidien (dans cet exposé)

Une sécurité prouvée ?

On ne connaît pas de construction de chiffrement
dont on peut prouver qu'elle est sûre

Solution : on fait reposer la sécurité sur la difficulté supposée de certains problèmes algorithmiques.

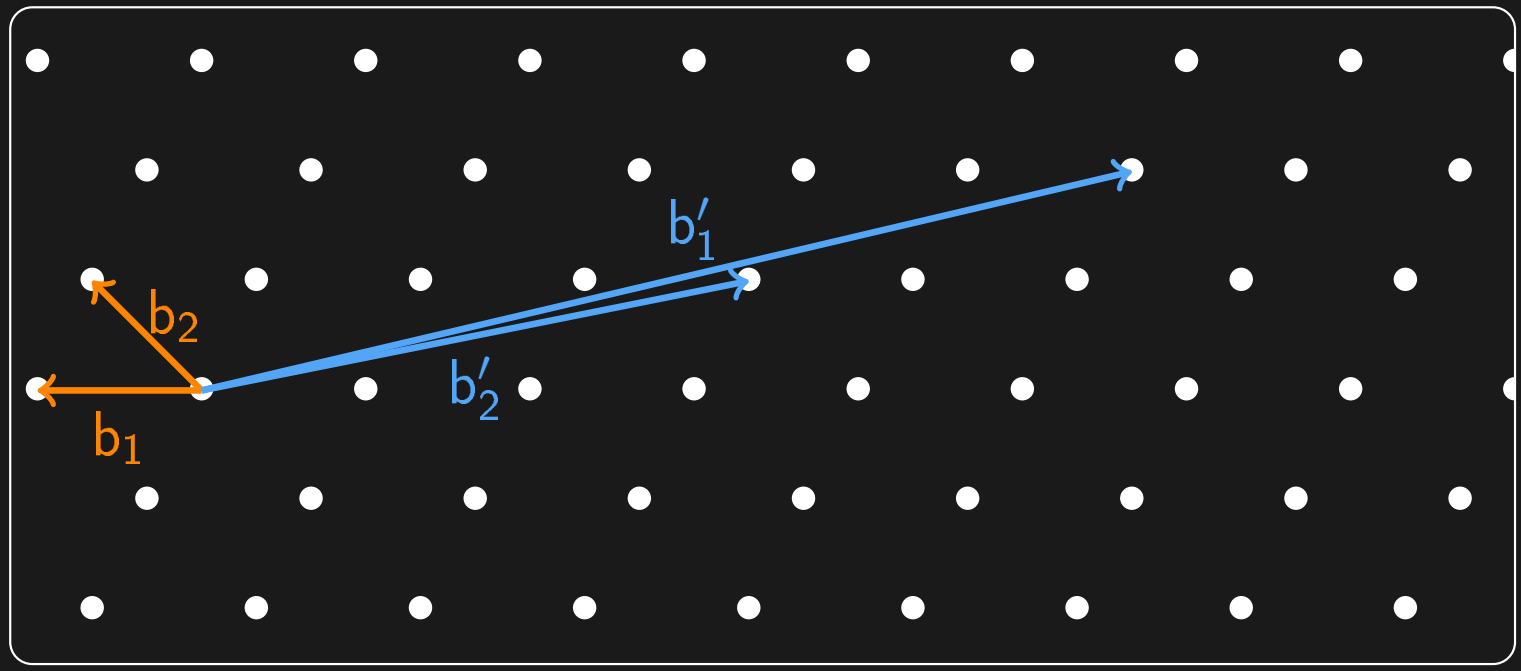
$\Rightarrow$ ce sont les "axiomes" de la cryptographie

Exemples : factorisation, logarithme discret, ...
décodage dans un réseau euclidien (dans cet exposé)

Formellement : un problème algorithmique est difficile s'il n'existe pas d'algorithme polynomial pour le résoudre

Réseaux euclidiens

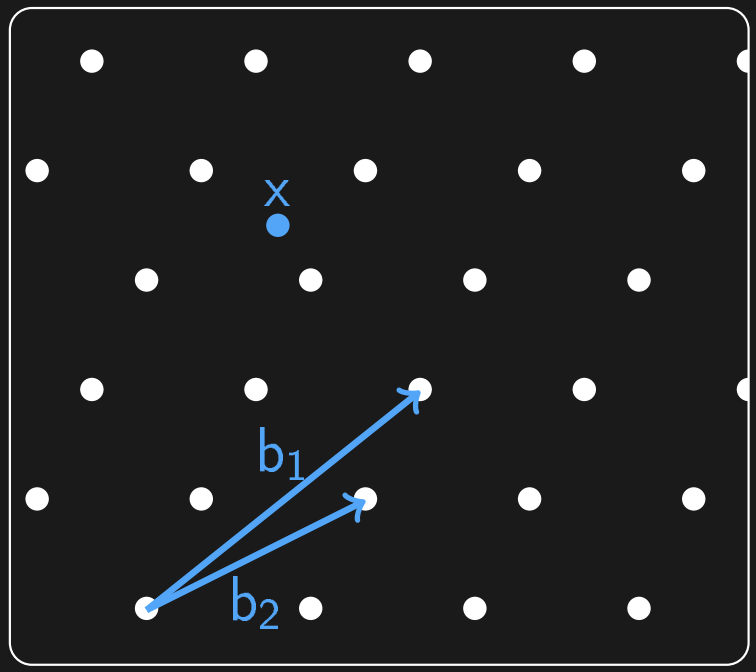
Réseaux



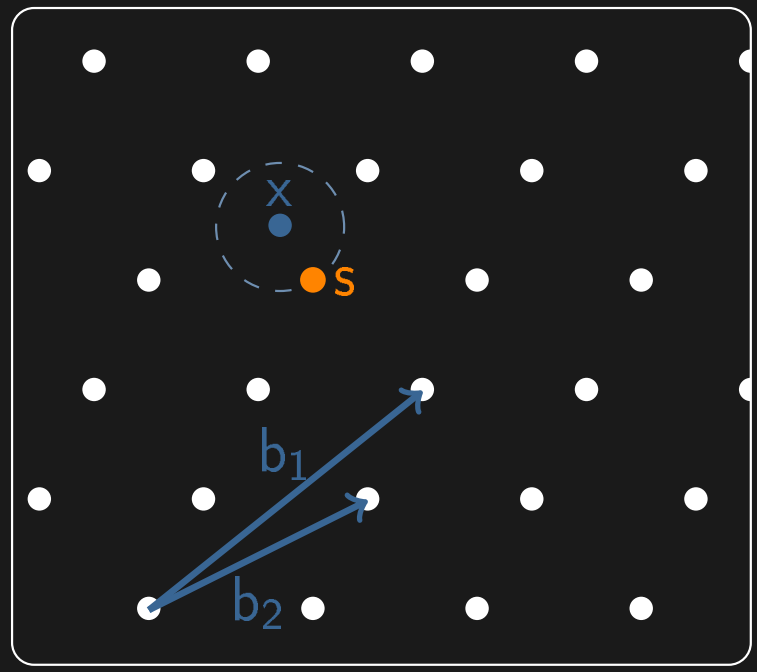
- $\mathcal{L} = \{\sum_{i=1}^{n} x_i b_i \mid \forall i,\ x_i \in \mathbb{Z}\}$ est un réseau
- $(b_1, \ldots, b_n) =: B \in \mathrm{GL}_n(\mathbb{R})$ est une base (non unique)
- n est la dimension du réseau

Décodage

Entrée:



Sortie:



Problème du décodage

$$\|s - x\| = \min_{v \in L} \|v - x\|$$

Difficulté

Le problème du décodage est

- facile si on a une base courte du réseau
  ⇝ algorithme polynomial pour décoder si on a une base courte
- difficile si on a une mauvaise base du réseau
  ⇝ pas d'algorithme polynomial connu pour décoder avec une mauvaise base

Difficulté

Le problème du décodage est

- facile si on a une base courte du réseau

  $\rightsquigarrow$ algorithme polynomial pour décoder si on a une base courte

- difficile si on a une mauvaise base du réseau

  $\rightsquigarrow$ pas d'algorithme polynomial connu pour décoder avec une mauvaise base

La difficulté augmente avec la dimension n

(polynomial = polynomial en n)

En pratique

Décodage avec une mauvaise base :

- $n = 2 \rightsquigarrow$ facile, très rapide en pratique

En pratique

Décodage avec une mauvaise base :

- $n = 2 \rightsquigarrow$ facile, très rapide en pratique
- jusqu'à $n = 60 \rightsquigarrow$ quelques minutes sur un ordi portable

En pratique

Décodage avec une mauvaise base :

- $n = 2 \rightsquigarrow$ facile, très rapide en pratique
- jusqu'à $n = 60 \rightsquigarrow$ quelques minutes sur un ordi portable
- jusqu'à $n = 180 \rightsquigarrow$ quelques jours sur un gros calculateur [DSW21]

[DSW21] Ducas, Stevens, van Woerden. Advanced Lattice Sieving on GPUs, with Tensor Cores.

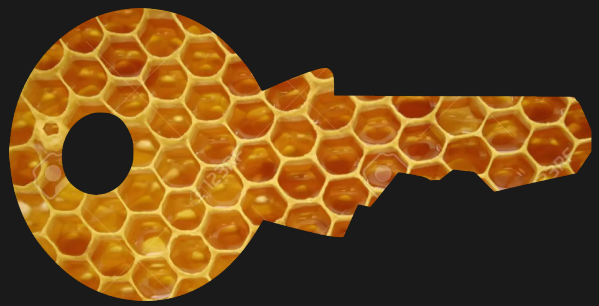
En pratique

Décodage avec une mauvaise base :

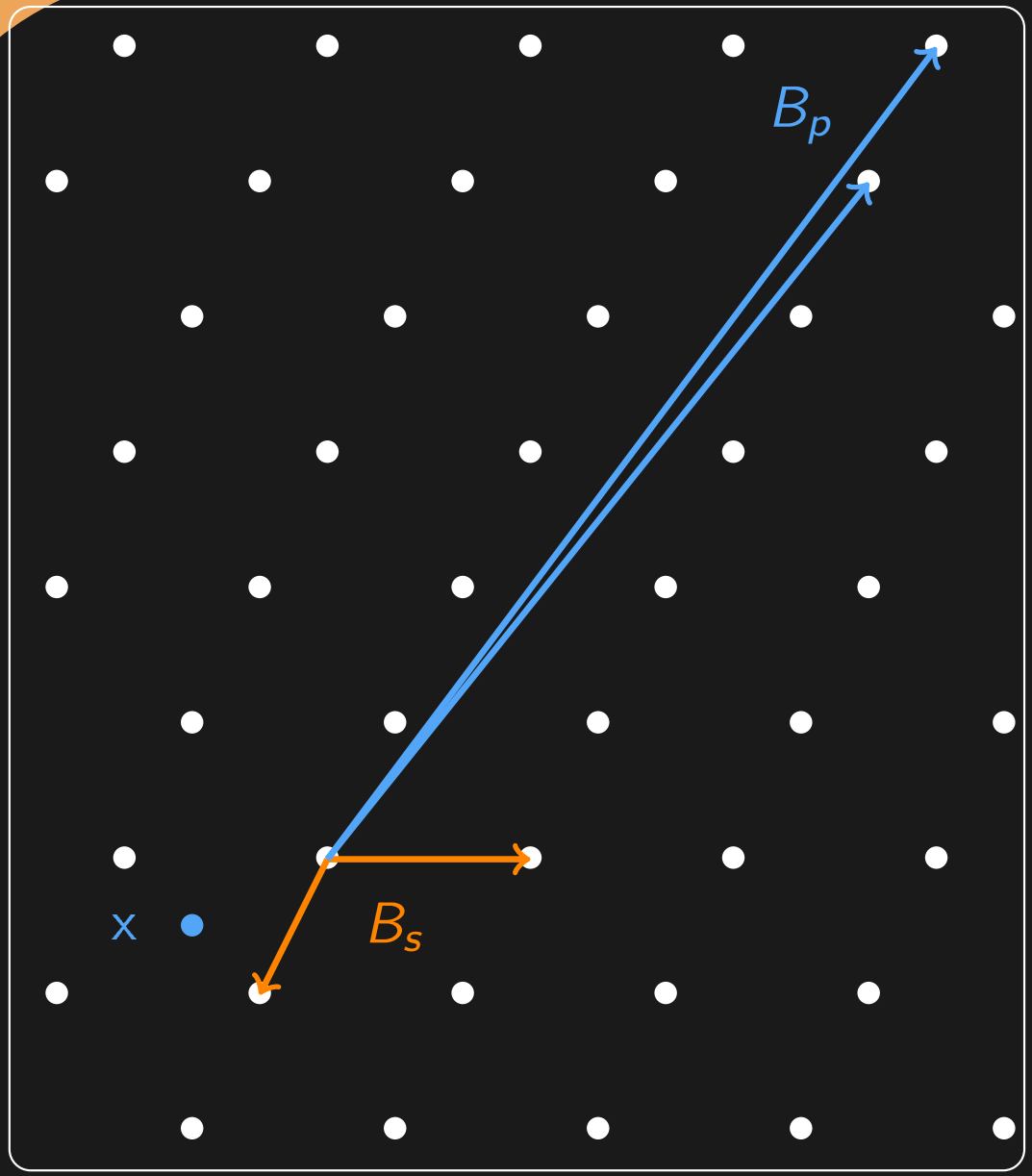
- $n = 2 \rightsquigarrow$ facile, très rapide en pratique
- jusqu'à $n = 60 \rightsquigarrow$ quelques minutes sur un ordi portable
- jusqu'à $n = 180 \rightsquigarrow$ quelques jours sur un gros calculateur [DSW21]
- entre $n = 500$ et $n = 1000 \rightsquigarrow$ cryptographie

[DSW21] Ducas, Stevens, van Woerden. Advanced Lattice Sieving on GPUs, with Tensor Cores.

Chiffrement à base de réseaux

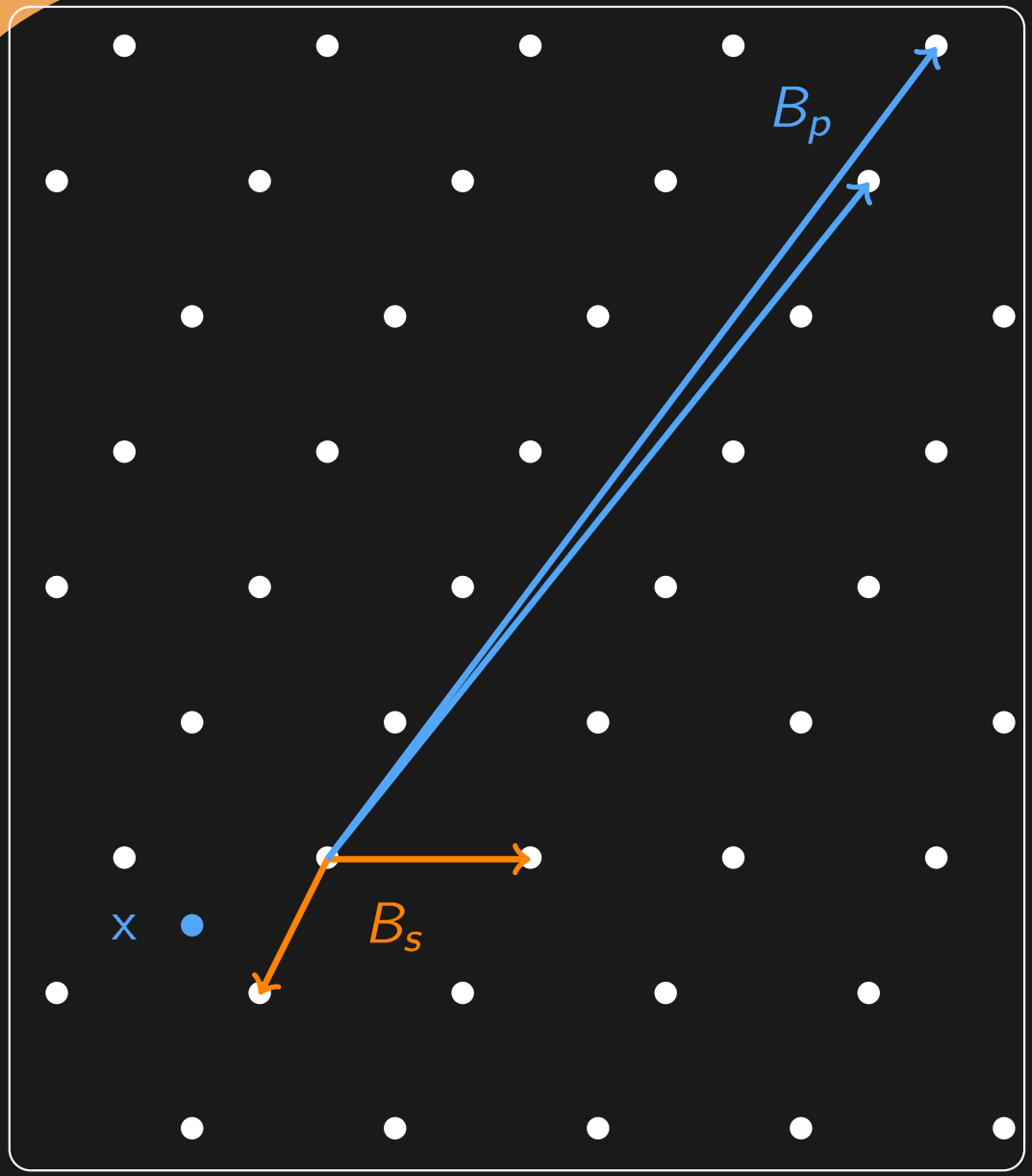


Chiffrement asymétrique utilisant des réseaux



$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

Chiffrement asymétrique utilisant des réseaux

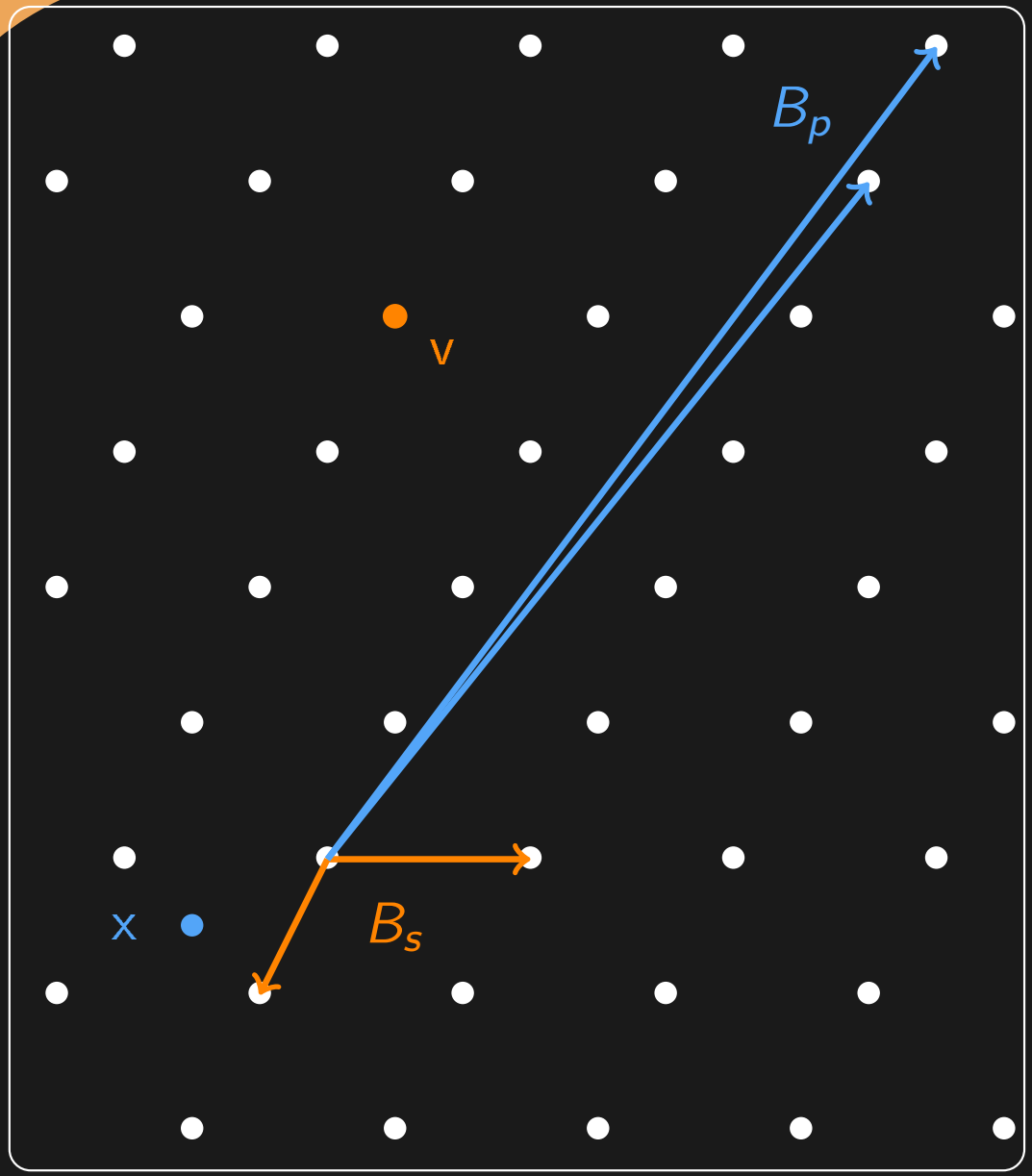


$\mathrm{pk} = (B_p, \mathrm{x})$

$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Chiffrement asymétrique utilisant des réseaux



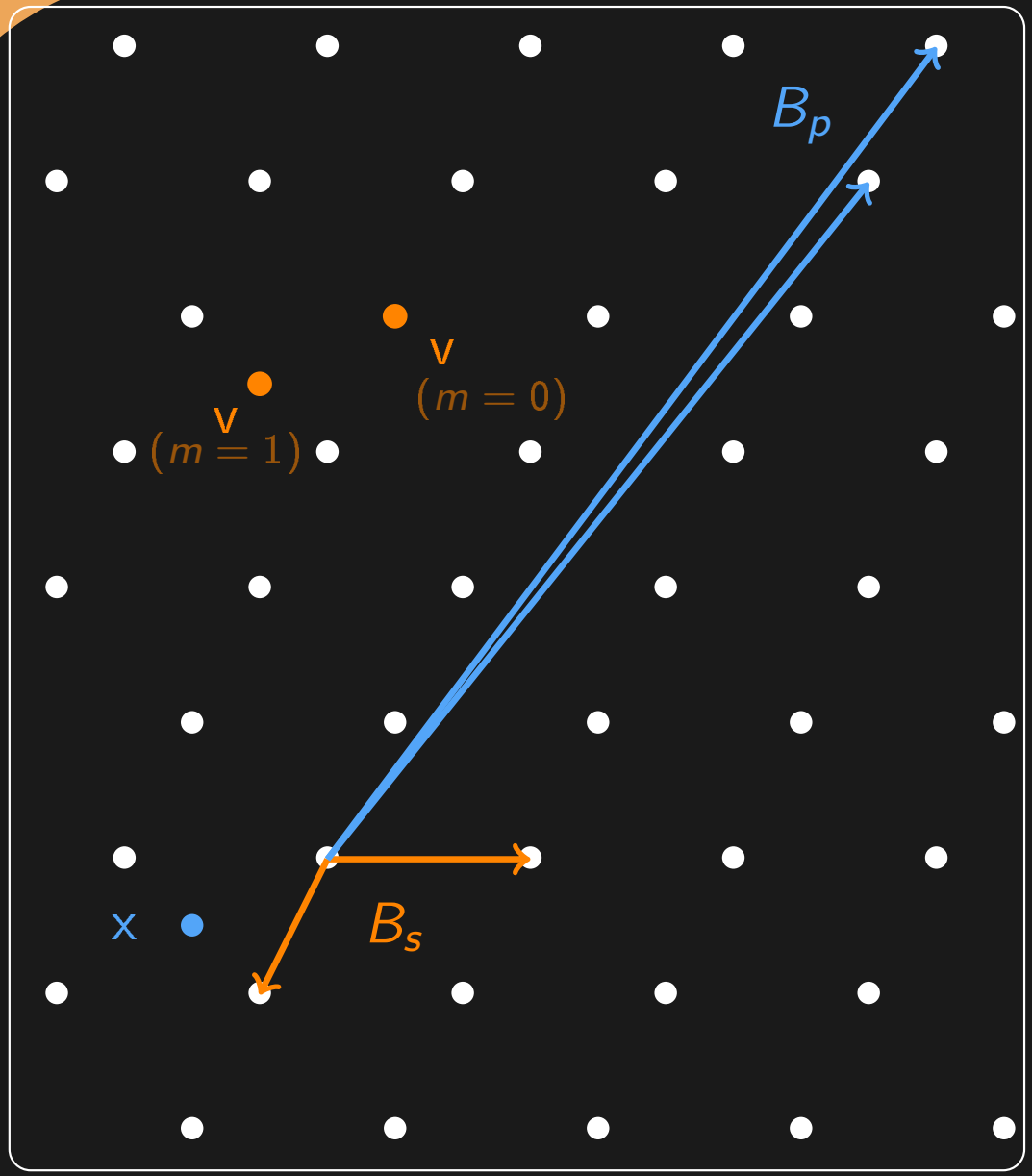
$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Enc(m, pk):

- tirer v $\in L$ au hasard

Chiffrement asymétrique utilisant des réseaux



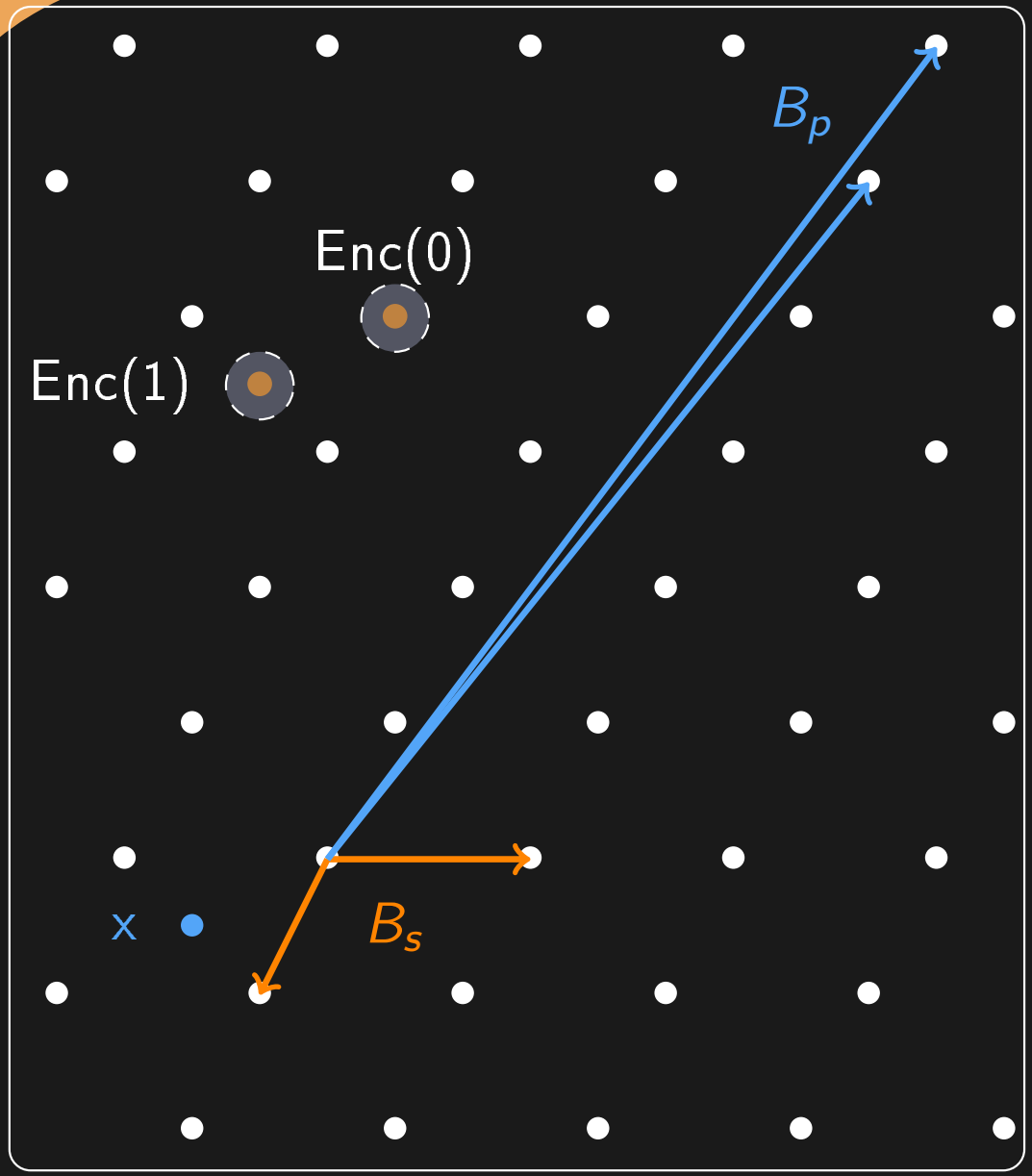
$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Enc(m, pk):

- tirer $\mathrm{v} \in L$ au hasard
- si $m = 1$: $\mathrm{v} \leftarrow \mathrm{v} + \mathrm{x}$

Chiffrement asymétrique utilisant des réseaux



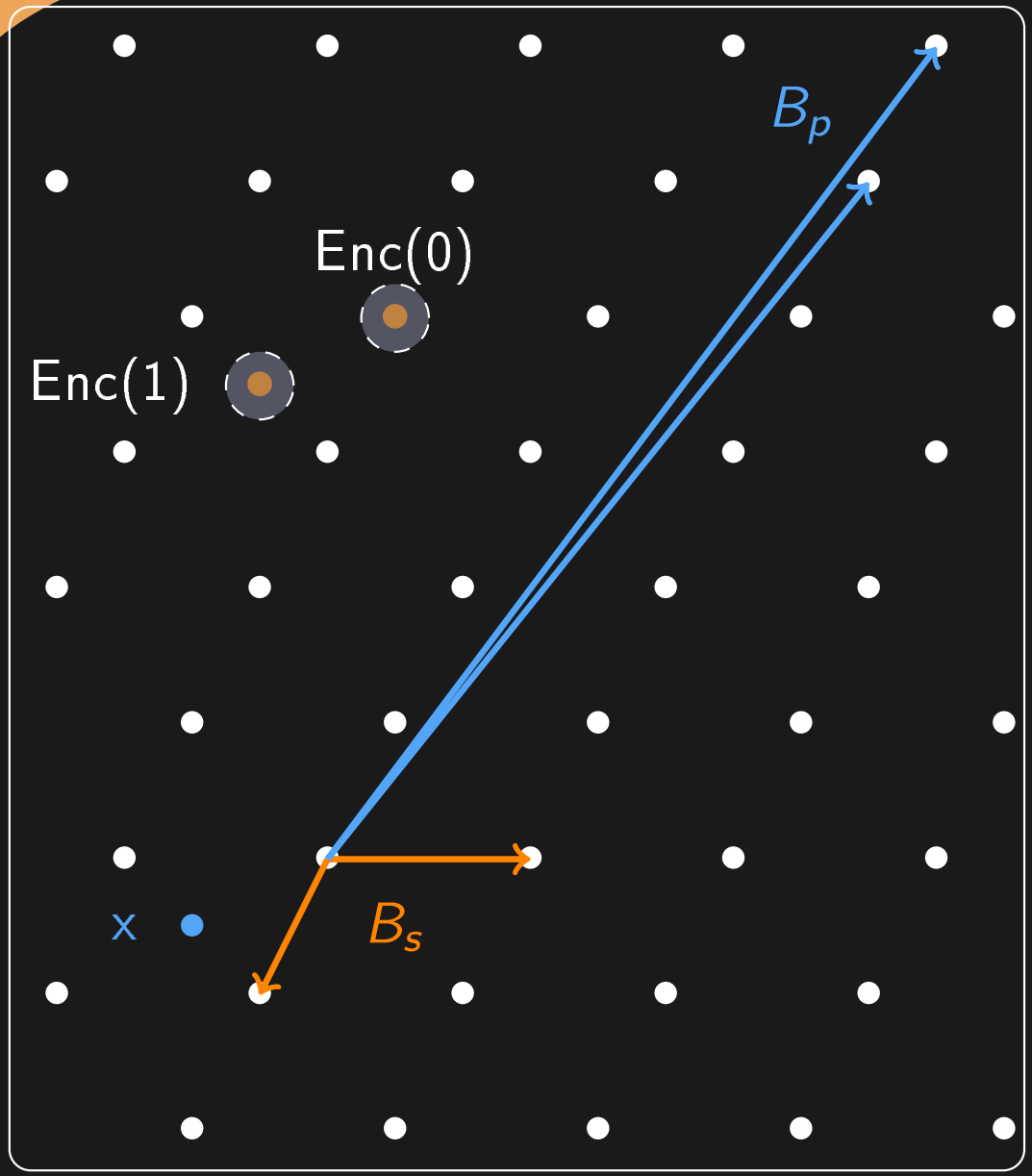
$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Enc(m, pk):

- tirer $\mathrm{v} \in L$ au hasard
- si $m = 1$: $\mathrm{v} \leftarrow \mathrm{v} + \mathrm{x}$
- tirer $\mathrm{e} \in \mathbb{R}^n$ petit
- renvoyer $\mathrm{c} = \mathrm{v} + \mathrm{e}$

Chiffrement asymétrique utilisant des réseaux



$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

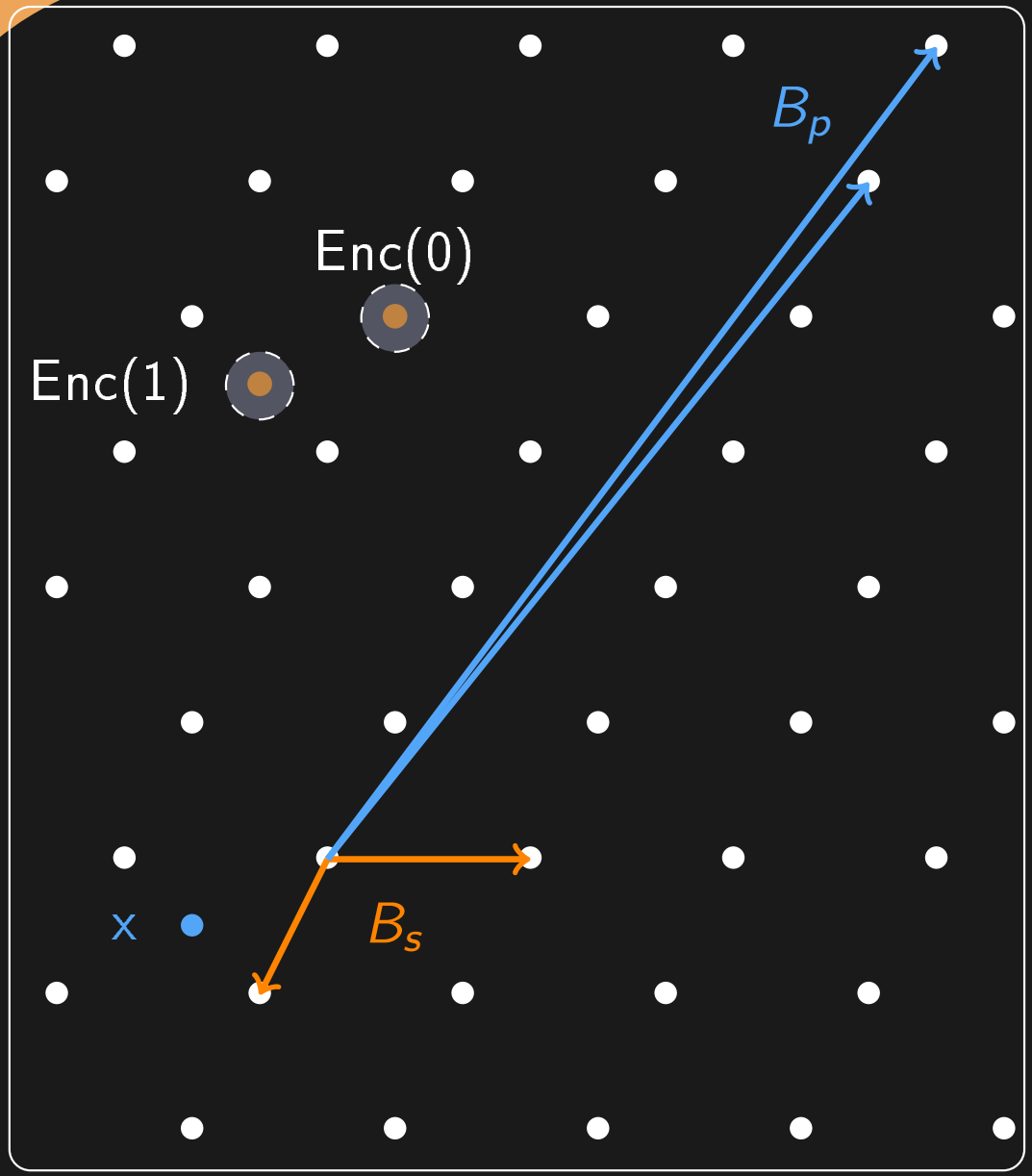
Enc(m, pk):

- tirer $\mathrm{v} \in L$ au hasard
- si $m = 1$: $\mathrm{v} \leftarrow \mathrm{v} + \mathrm{x}$
- tirer $\mathrm{e} \in \mathbb{R}^n$ petit
- renvoyer $\mathrm{c} = \mathrm{v} + \mathrm{e}$

Dec(c, sk):

- $\mathrm{s} \leftarrow \mathrm{Decoder}(\mathrm{c})$
- si $\|\mathrm{s} - \mathrm{c}\|$ petit $\rightsquigarrow m = 0$
- sinon $\rightsquigarrow m = 1$

Chiffrement asymétrique utilisant des réseaux

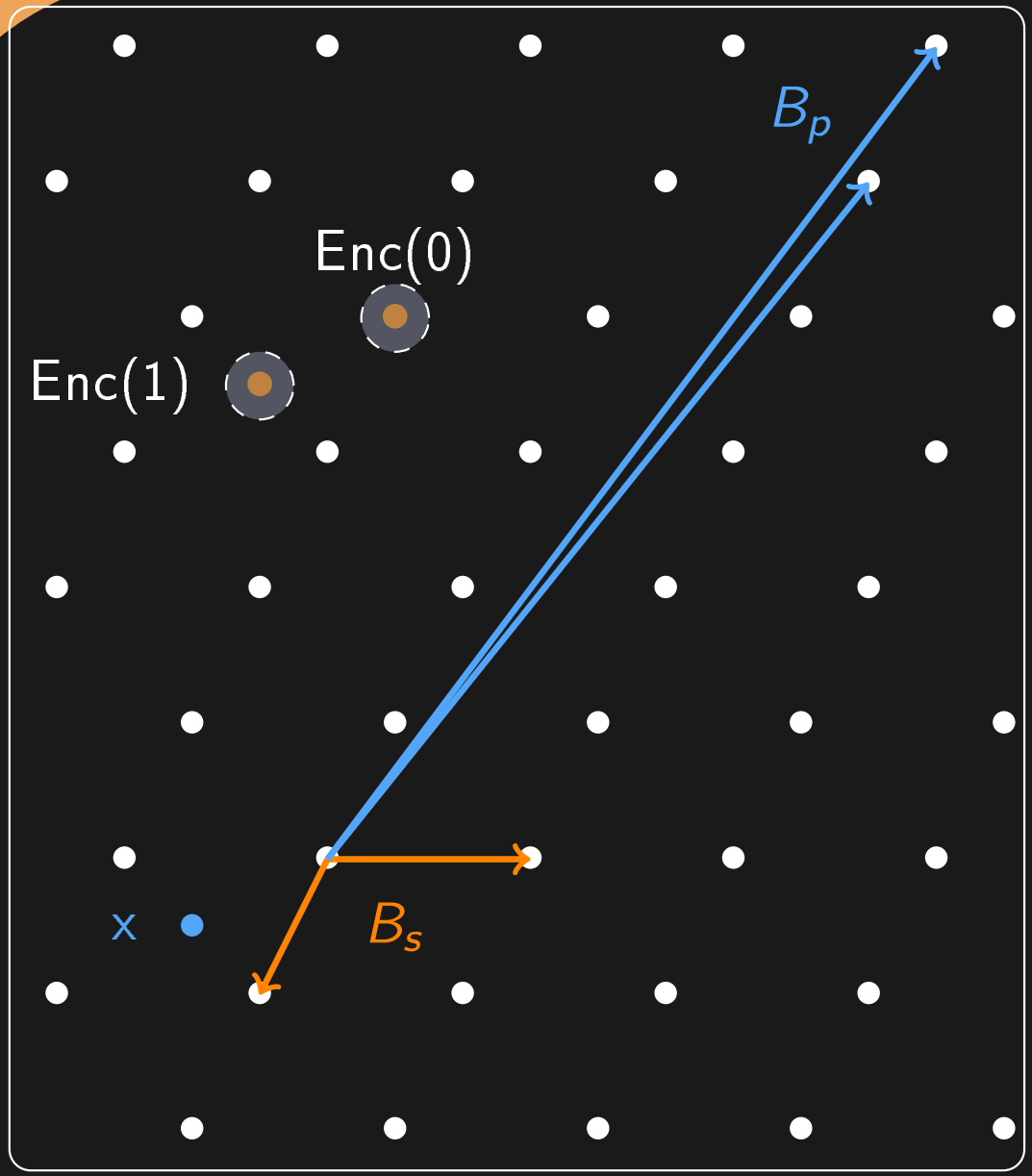


$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Correction : ✓
(Enc(1) est loin du réseau
Enc(0) est proche du réseau)

Chiffrement asymétrique utilisant des réseaux



$\mathrm{pk} = (B_p, \mathrm{x})$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Correction : ✓
(Enc(1) est loin du réseau
Enc(0) est proche du réseau)

Sécurité : ✓
(S'il existe un attaquant, alors le problème du décodage avec une mauvaise base est facile)

Conclusion

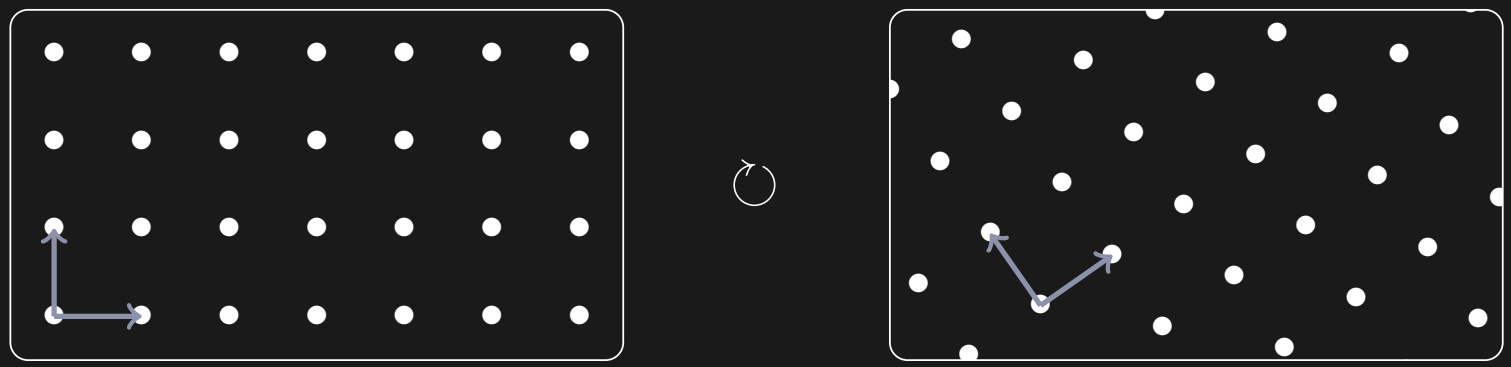
Ma recherche : faire le tri dans les problèmes supposés difficiles

(détecter ceux qui sont en fait faciles, trier les autres pas ordre de difficulté, . . .)

Conclusion

Ma recherche : faire le tri dans les problèmes supposés difficiles

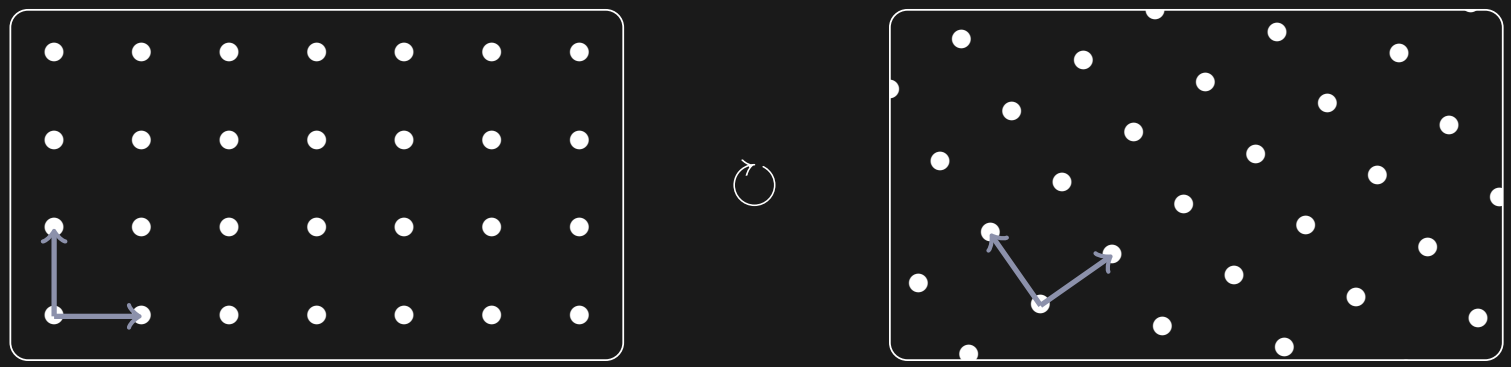
(détecter ceux qui sont en fait faciles, trier les autres pas ordre de difficulté, . . .)



Conclusion

Ma recherche : faire le tri dans les problèmes supposés difficiles

(détecter ceux qui sont en fait faciles, trier les autres pas ordre de difficulté, . . .)



Merci de votre attention