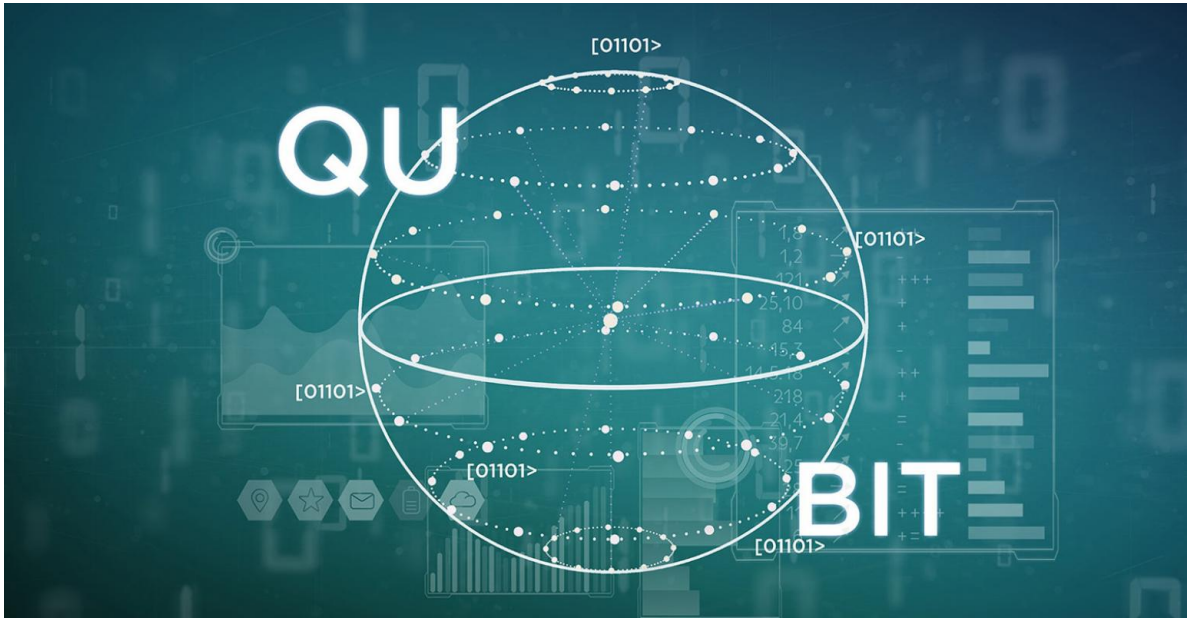
Attaques pratiques par réduction de réseaux, assistées par canaux auxiliaires

Thibauld Feneuil

Master Parisien de Recherche en Informatique
et Télécom Paris

Mercredi 9 Septembre 2020

Une menace pour la cryptographie contemporaine

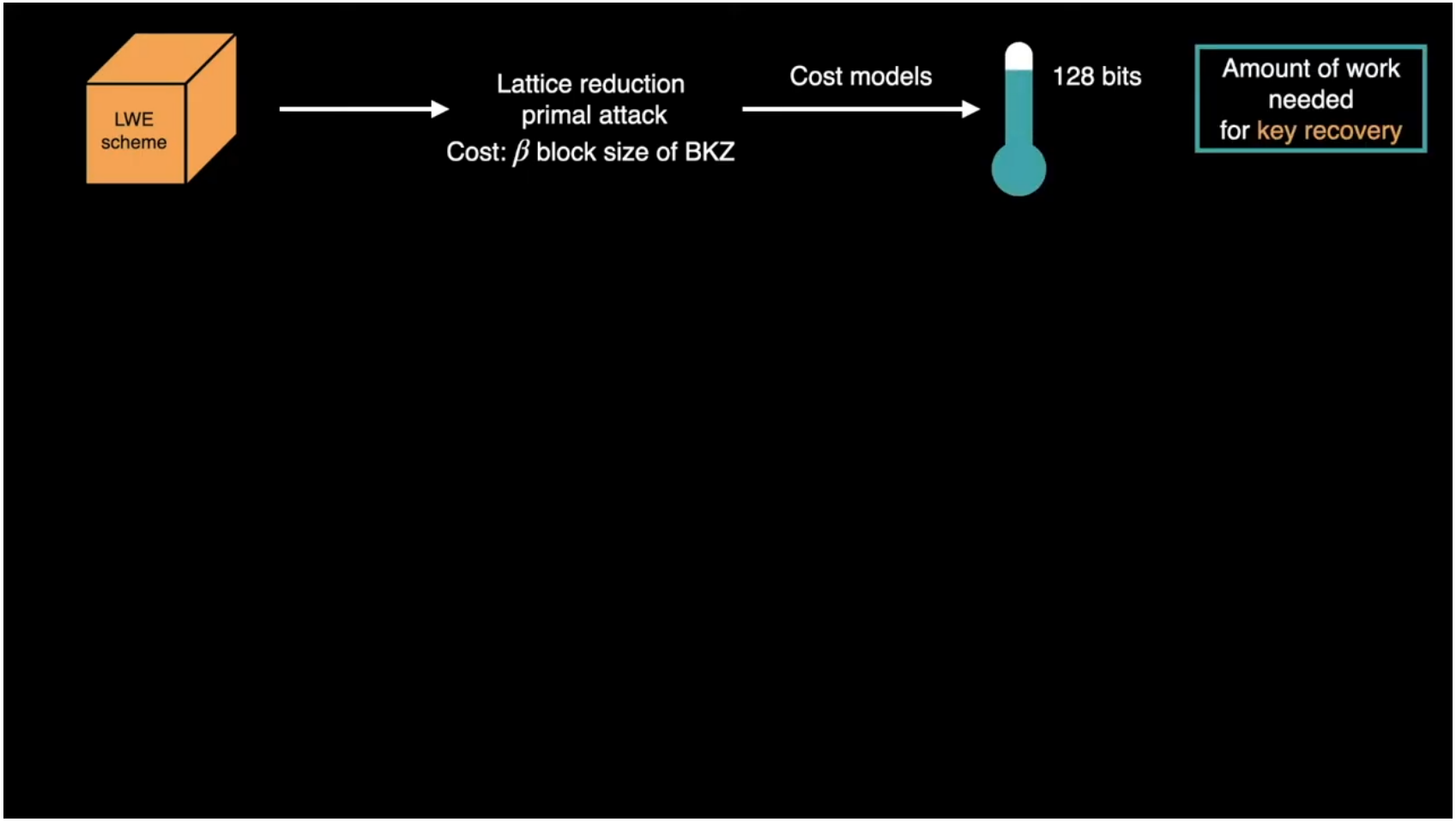


- *Source : https ://www.machinedesign.com/*

National Institute of Standards and Technology
U.S. Department of Commerce

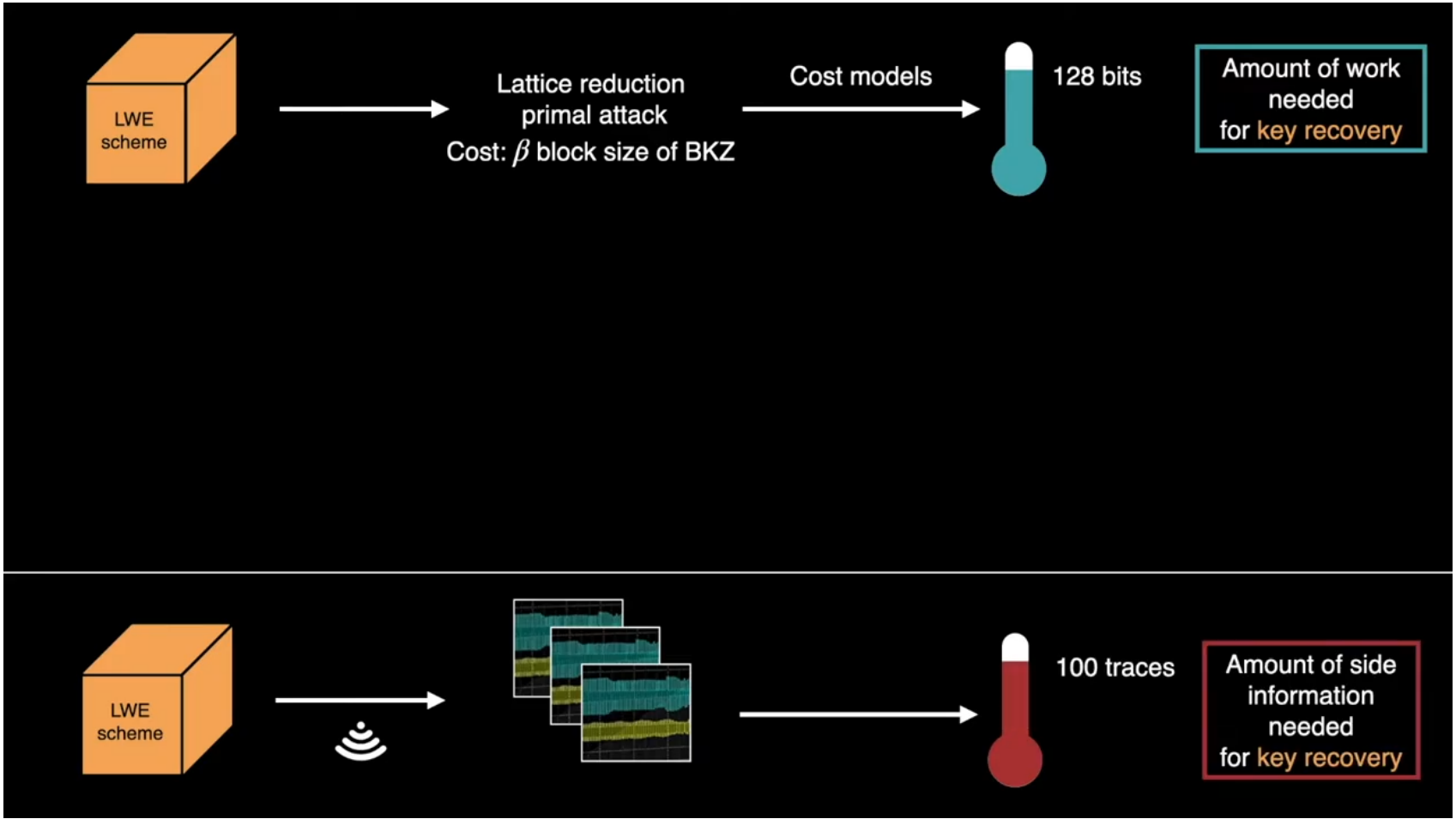
Projet « *Post-quantum cryptography* »

Positionnement des travaux



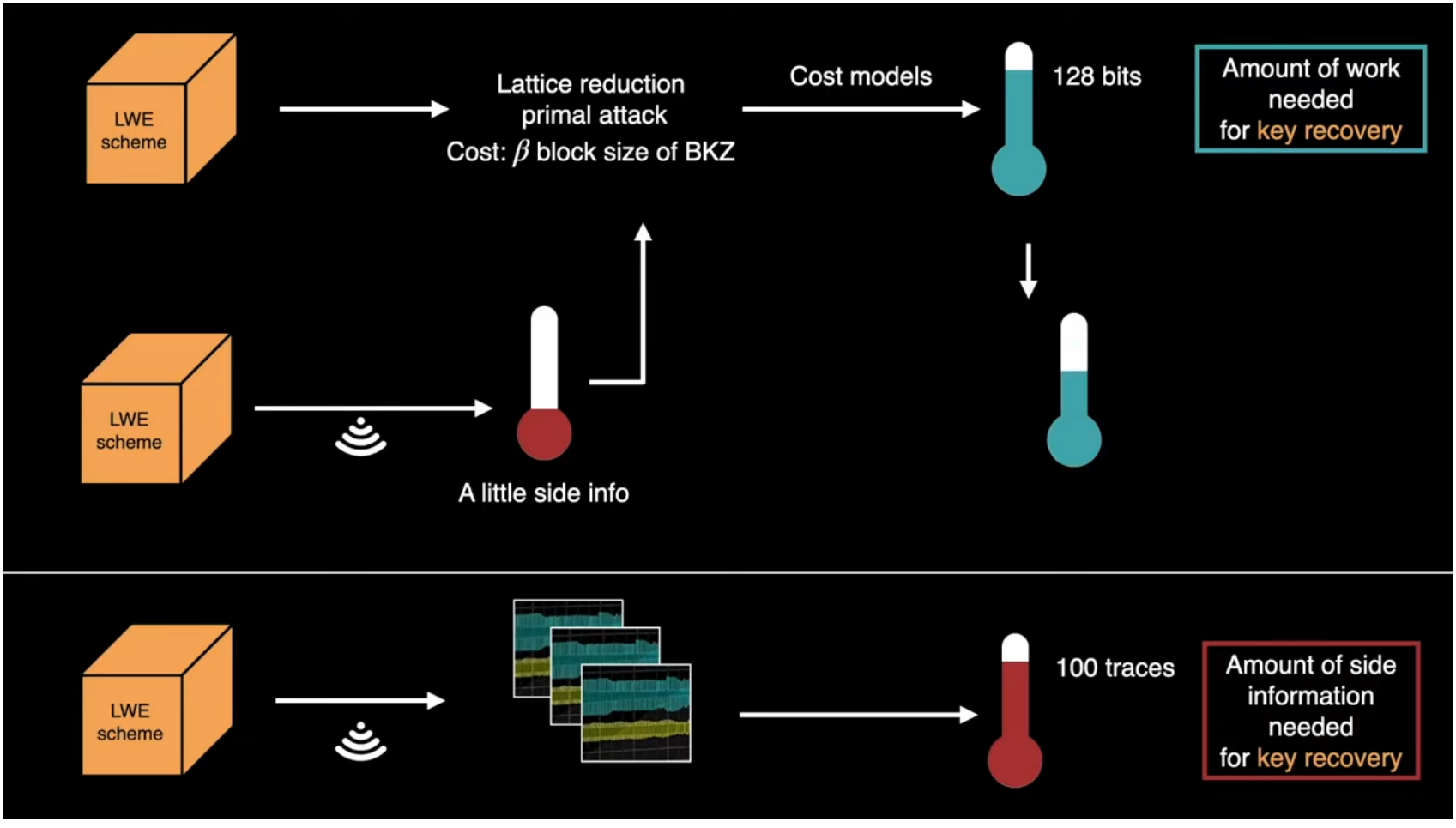
- *Image issue de la Conférence Crypto2020.*
LWE with Side Information : Attacks and Concrete Security Estimation

Positionnement des travaux



- *Image issue de la Conférence Crypto2020. LWE with Side Information : Attacks and Concrete Security Estimation*

Positionnement des travaux



- *Image issue de la Conférence Crypto2020.*
LWE with Side Information : Attacks and Concrete Security Estimation

Table des matières

Table des matières

Les réseaux euclidiens, *qu'es aquò ?*

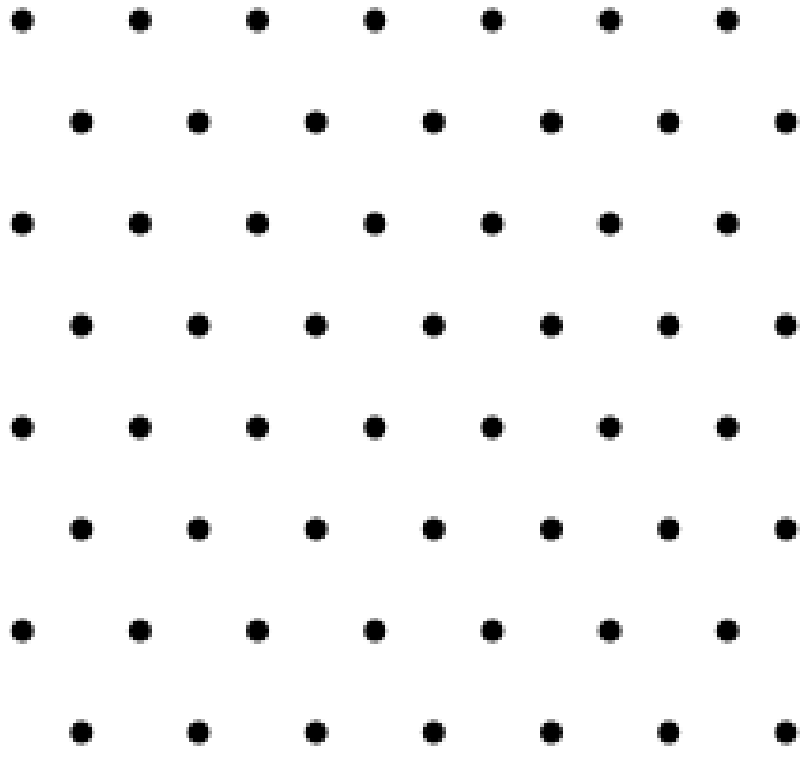


FIGURE – Un réseau de dimension 2

Les réseaux euclidiens, *qu'es aquò ?*

Réseau euclidien

Sous-groupe discret additif de $\mathbb{R}^m$.

Les réseaux euclidiens, *qu'es aquò ?*

Réseau euclidien

Sous-groupe discret additif de $\mathbb{R}^m$.

Réseau euclidien (équivalent)

Ensemble des combinaisons linéaires à coefficients **entiers** de n vecteurs $\{\mathbf{b}_j\} \subset \mathbb{R}^m$ linéairement indépendants :

$$\Lambda := \left\{ \sum_j z_j \mathbf{b}_j : z_j \in \mathbb{Z} \right\}.$$

Nous appelons m la *dimension* de Λ et n son *rang*.

Un problème difficile sur les réseaux

Learning With Errors (LWE)

Soit une paire $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{b} = \mathbf{Az} + \mathbf{e} \in \mathbb{Z}_q^n)$ où

- $\mathbf{A} \leftarrow_\$ \mathcal{M}_{n,m}(\mathbb{Z}_q)$,
- $\begin{cases} \mathbf{z} \leftarrow_\$ \chi^m \\ \mathbf{e} \leftarrow_\$ \chi^n \end{cases}$, avec χ une petite distribution.

Trouver $\mathbf{z}$.

Un problème difficile sur les réseaux

Learning With Errors (LWE)

Soit une paire $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{b} = \mathbf{Az} + \mathbf{e} \in \mathbb{Z}_q^n)$ où

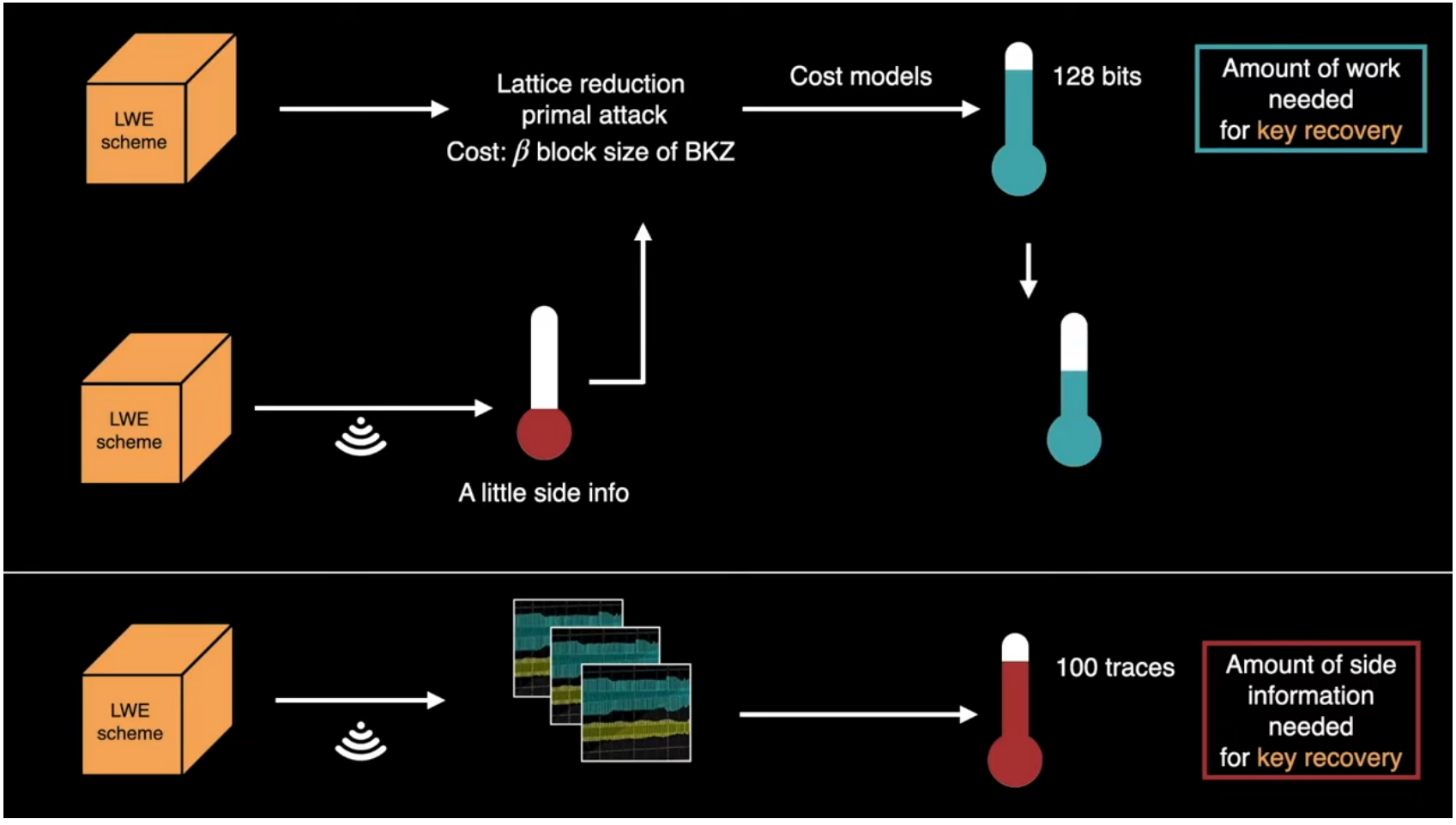
- $\mathbf{A} \leftarrow_\$ \mathcal{M}_{n,m}(\mathbb{Z}_q)$,
- $\begin{cases} \mathbf{z} \leftarrow_\$ \chi^m \\ \mathbf{e} \leftarrow_\$ \chi^n \end{cases}$, avec χ une petite distribution.

Trouver $\mathbf{z}$.

☞ Résoudre ce problème, pour certains choix de paramètres, est supposé difficile même avec l'usage de l'informatique quantique.

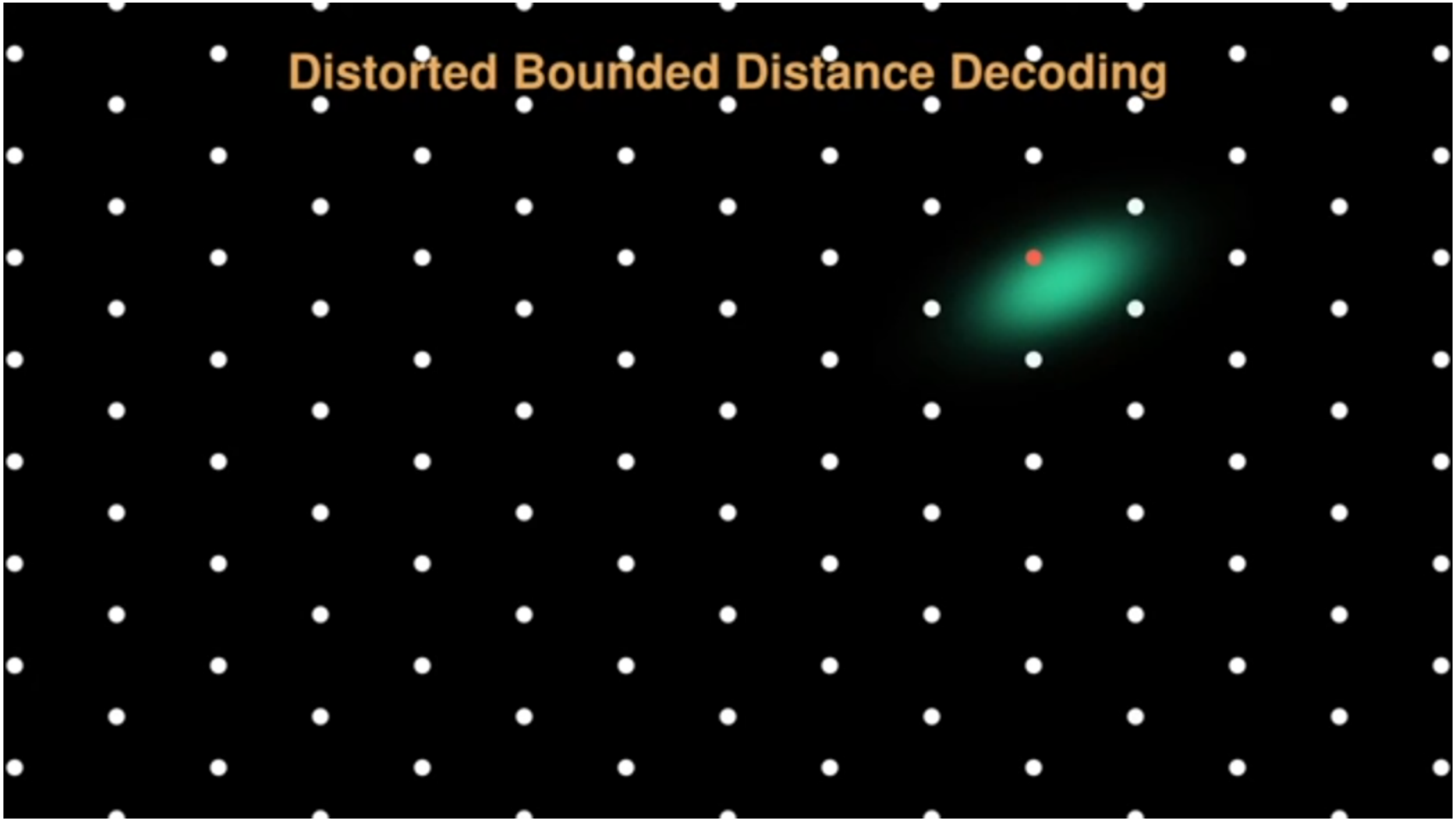
Table des matières

Le framework *Leaky-LWE-Estimator*



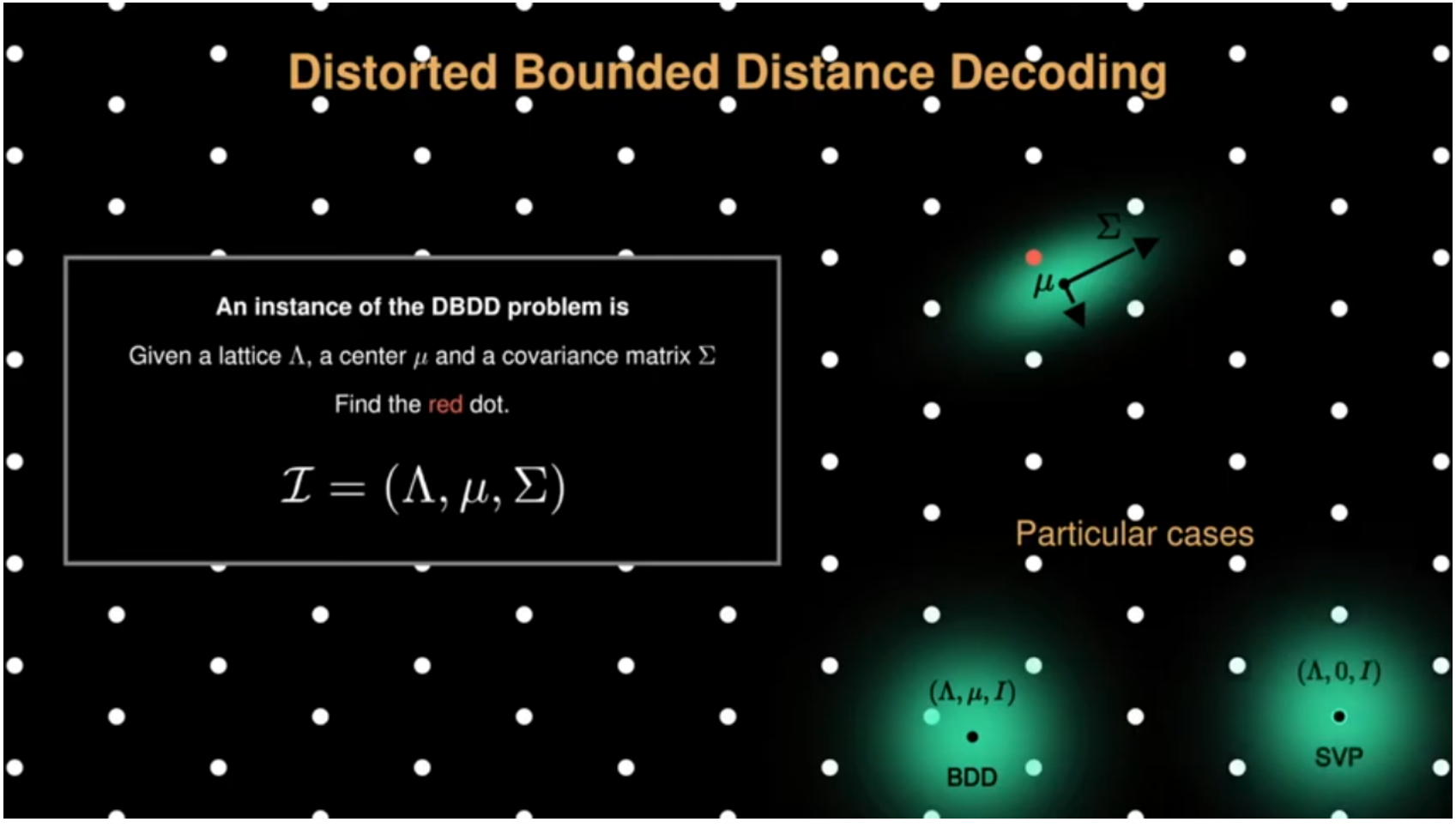
- *Image issue de la Conférence Crypto2020. LWE with Side Information : Attacks and Concrete Security Estimation*

Distorted Bounded Distance Decoding (DBDD)



- Image issue de la Conférence Crypto2020.
LWE with Side Information : Attacks and Concrete Security Estimation

Distorted Bounded Distance Decoding (DBDD)



- Image issue de la Conférence Crypto2020.
LWE with Side Information : Attacks and Concrete Security Estimation

Exploiter des indices dans l'attaque primale

Méthodologie de *Leaky-LWE-Estimator*

$$\mathsf{LWE/BDD} \longrightarrow \mathsf{DBDD}_{\Lambda_0,\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0} \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_1,\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1} \; \vdots \; \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_h,\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h} \longrightarrow \mathsf{uSVP}_{\Lambda'} \longrightarrow \text{Réduction de réseaux}$$

Exploiter des indices dans l'attaque primale

Méthodologie de *Leaky-LWE-Estimator*

$$\mathsf{LWE/BDD} \longrightarrow \mathsf{DBDD}_{\Lambda_0,\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0} \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_1,\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1} \vdots \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_h,\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h} \longrightarrow \mathsf{uSVP}_{\Lambda'} \longrightarrow \text{Réduction de réseaux}$$

Intégration des indices :

Modification d'une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Exploiter des indices dans l'attaque primale

Méthodologie de *Leaky-LWE-Estimator*

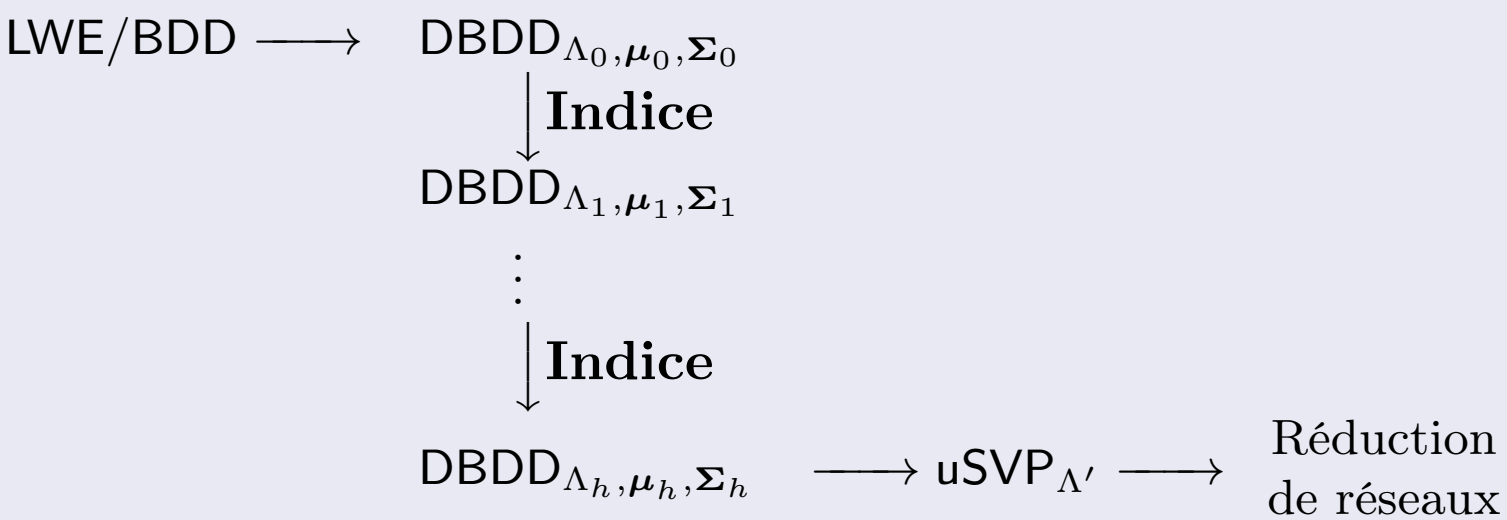
$$\mathsf{LWE/BDD} \longrightarrow \mathsf{DBDD}_{\Lambda_0,\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0} \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_1,\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1} \; \vdots \; \downarrow \textbf{Indice} \; \mathsf{DBDD}_{\Lambda_h,\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h} \longrightarrow \mathsf{uSVP}_{\Lambda'} \longrightarrow$$ Réduction de réseaux

Transformation en uSVP :

Le plus court vecteur est $\bar{\mathbf{s}} := (\mathbf{e}, \mathbf{z}, 1)$
en notant $(\mathbf{A}, \mathbf{b} = \mathbf{Az} + \mathbf{e})$ le problème LWE initial.

Exploiter des indices dans l'attaque primale

Méthodologie de *Leaky-LWE-Estimator*



Réduction de réseau :

Algorithme β-BKZ : Base de $\Lambda \mapsto$ Vecteur court de Λ

Le β **minimal nécessaire** sera notre unité de mesure pour évaluer la sécurité des cryptosystèmes, nous appelerons *bikz* l'unité correspondante.

Intégration des indices

Quatre types d'indices

- Indices parfaits : $\langle \mathbf{s}, \mathbf{v} \rangle = l$
- Indices modulaires : $\langle \mathbf{s}, \mathbf{v} \rangle \equiv l \pmod{k}$
- Indices bruités/approximés : $\langle \mathbf{s}, \mathbf{v} \rangle = l + \epsilon_\sigma$
- Vecteurs courts : $\mathbf{v} \in \Lambda$

☞ Chaque indice vont affecter le réseau Λ, le vecteur μ et la matrice de covariance de $\mathbf{\Sigma}$ d'une manière totalement prévisible.

Intégration des indices, formulaire

Type d'indices	**Parfait**	**Modulaire**
Formalisme	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{k}$
$\boldsymbol{\Lambda'}$	$\Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle = 0\}$	$\Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{k}\}$
$\boldsymbol{\mu'}$	$\boldsymbol{\mu} - \frac{\langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \cdot \boldsymbol{\Sigma} \bar{\mathbf{v}}$	$\boldsymbol{\mu}$
$\boldsymbol{\Sigma'}$	$\boldsymbol{\Sigma} - \frac{(\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}}$	$\boldsymbol{\Sigma}$

Type d'indices	**Bruité**	**Approximé**
Formalisme	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle + \epsilon_{\sigma_e} = 0$	$\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \sim \mathcal{D}_{ap}$
$\boldsymbol{\Lambda'}$	Λ	Λ
$\boldsymbol{\mu'}$	$\boldsymbol{\mu} - \frac{\langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2} \cdot \boldsymbol{\Sigma} \bar{\mathbf{v}}$	$\Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|^2}$
$\boldsymbol{\Sigma'}$	$\boldsymbol{\Sigma} - \frac{(\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}} + \sigma_e^2}$	$\Pi_{\bar{\mathbf{v}}}^{\perp} \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}}^{\perp})^T + \sigma_{ap}^2 \cdot \frac{\bar{\mathbf{v}} \bar{\mathbf{v}}^T}{\|\bar{\mathbf{v}}\|^4}$

Table des matières

Réduction de dimension

Prenons un **indice parfait** « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ » et étudions-le...

Réduction de dimension

Prenons un **indice parfait** « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ » et étudions-le...

☞ Une instance LWE est un système d'équations q-modulaires :

$$\mathbf{b} \equiv \mathbf{A}\mathbf{z} + \mathbf{e} \pmod{q}$$

Réduction de dimension

Prenons un **indice parfait** « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ » et étudions-le...

☞ Une instance LWE est un système d'équations q-modulaires :

$$\mathbf{b} \equiv \mathbf{A}\mathbf{z} + \mathbf{e} \pmod{q}$$

Nous pouvons utiliser $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ pour **retirer une équation**, et avec la transformation en DBDD, nous obtenons un Λ' tel que

$$\mathrm{rank}(\Lambda') = \mathrm{rank}(\Lambda) - 1 \quad \text{et} \quad \dim(\Lambda') = \dim(\Lambda) - 1.$$

Réduction de dimension

Prenons un **indice parfait** « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ » et étudions-le...

☞ Une instance LWE est un système d'équations q-modulaires :

$$\mathbf{b} \equiv \mathbf{A}\mathbf{z} + \mathbf{e} \pmod{q}$$

Nous pouvons utiliser $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ pour **retirer une équation**, et avec la transformation en DBDD, nous obtenons un Λ' tel que

$$\mathrm{rank}(\Lambda') = \mathrm{rank}(\Lambda) - 1 \quad \text{et} \quad \dim(\Lambda') = \dim(\Lambda) - 1.$$

☞ **Dans le framework**, l'indice modifie le réseau avec

$$\Lambda' \leftarrow \Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle = 0\}.$$

Réduction de dimension

Prenons un **indice parfait** « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ » et étudions-le...

☞ Une instance LWE est un système d'équations q-modulaires :

$$\mathbf{b} \equiv \mathbf{A}\mathbf{z} + \mathbf{e} \pmod{q}$$

Nous pouvons utiliser $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ pour **retirer une équation**, et avec la transformation en DBDD, nous obtenons un Λ' tel que

$$\mathrm{rank}(\Lambda') = \mathrm{rank}(\Lambda) - 1 \quad \text{et} \quad \dim(\Lambda') = \dim(\Lambda) - 1.$$

☞ **Dans le framework**, l'indice modifie le réseau avec

$$\Lambda' \leftarrow \Lambda \cap \{\mathbf{x} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle = 0\}.$$

Nous avons $\mathrm{rank}(\Lambda') = \mathrm{rank}(\Lambda) - 1$ et $\dim(\Lambda') = \dim(\Lambda)$.

Réduction de dimension

Réduction de dimension

Soit une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Il est possible de transformer $\mathcal{I}$ en une instance $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ dont le réseau est de rang maximal.

Si nous avons $\boldsymbol{\Gamma}$ tel que $\Lambda = \boldsymbol{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\text{rank}(\Lambda) = \text{rank}(\Lambda')$, alors

$$\boldsymbol{\mu}' \leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\mu},$$
$$\boldsymbol{\Sigma}' \leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1},$$

Réduction de dimension

Réduction de dimension

Soit une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Il est possible de transformer $\mathcal{I}$ en une instance $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ dont le réseau est de rang maximal.

Si nous avons $\boldsymbol{\Gamma}$ tel que $\Lambda = \boldsymbol{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\text{rank}(\Lambda) = \text{rank}(\Lambda')$, alors

$$\boldsymbol{\mu}' \leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\mu},$$
$$\boldsymbol{\Sigma}' \leftarrow (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1},$$

☞ À présent, **les instances DBDD manipulées ont toutes leur réseau de rang maximal**.

Amélioration des performances

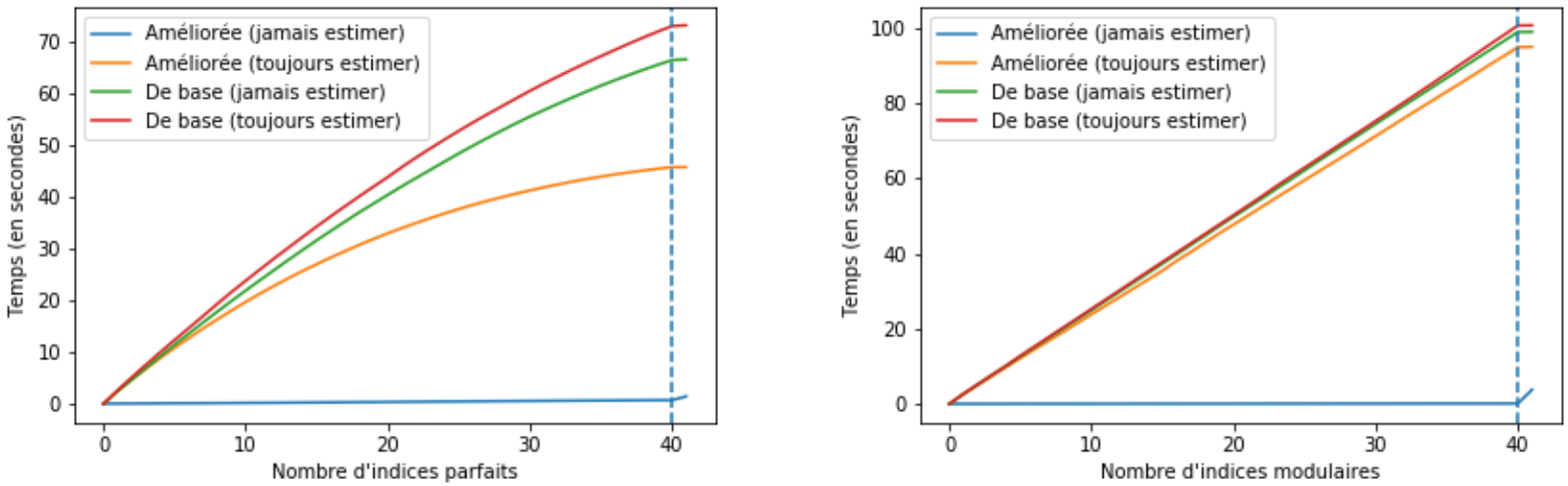


FIGURE – Comparaison de la vitesse de la version de base avec celle de la version améliorée de *Leaky-LWE-Estimator*.

Intégration des indices approximés

☞ Étudions à présent les indices approximés...

Intégration des indices approximés

☞ Étudions à présent les indices approximés...

Effet sur l'instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Nous connaissons la distribution *a posteriori* $\mathcal{D}_{ap}$ de $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$, de moyenne μ_{ap} et de variance σ_{ap}. Pour prendre en compte cet indice, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda, \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\boldsymbol{\mu}' = \Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|^2}$$

$$\boldsymbol{\Sigma}' = (\Pi_{\bar{\mathbf{v}}}^{\perp}) \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}}^{\perp})^T + \sigma_{ap}^2 \cdot \frac{\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\|\bar{\mathbf{v}}\|^4}$$

Intégration des indices approximés

☞ Étudions à présent les indices approximés...

Effet sur l'instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Nous connaissons la distribution *a posteriori* $\mathcal{D}_{ap}$ de $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$, de moyenne μ_{ap} et de variance σ_{ap}. Pour prendre en compte cet indice, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda, \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\boldsymbol{\mu}' = \Pi_{\bar{\mathbf{v}}}^{\perp} \boldsymbol{\mu} + \mu_{ap} \cdot \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|^2}$$

$$\boldsymbol{\Sigma}' = (\Pi_{\bar{\mathbf{v}}}^{\perp}) \cdot \boldsymbol{\Sigma} \cdot (\Pi_{\bar{\mathbf{v}}}^{\perp})^T + \sigma_{ap}^2 \cdot \frac{\bar{\mathbf{v}}\bar{\mathbf{v}}^T}{\|\bar{\mathbf{v}}\|^4}$$

Problème...

Il s'est révélé par la suite que ces formules sont valides **uniquement dans le cas où le vecteur $\bar{\mathbf{v}}$ est un vecteur propre de $\boldsymbol{\Sigma}$.**

Intégration des indices approximés

Effet sur l'instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, *version corrigée*

Nous connaissons la distribution *a posteriori* $\mathcal{D}_{ap}$ de $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$, de moyenne μ_{ap} et de variance σ_{ap}. Pour prendre en compte cet indice, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda, \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \frac{\mu_{ap} - \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \boldsymbol{\Sigma} \bar{\mathbf{v}}$$

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} + \left(\frac{\sigma_{ap}^2}{(\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}})^2} - \frac{1}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \right) (\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T$$

Intégration des indices approximés

Effet sur l'instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, *version corrigée*

Nous connaissons la distribution *a posteriori* $\mathcal{D}_{ap}$ de $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle$, de moyenne μ_{ap} et de variance σ_{ap}. Pour prendre en compte cet indice, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda, \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \frac{\mu_{ap} - \langle \bar{\mathbf{v}}, \boldsymbol{\mu} \rangle}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \boldsymbol{\Sigma} \bar{\mathbf{v}}$$

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} + \left(\frac{\sigma_{ap}^2}{(\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}})^2} - \frac{1}{\bar{\mathbf{v}}^T \boldsymbol{\Sigma} \bar{\mathbf{v}}} \right) (\boldsymbol{\Sigma} \bar{\mathbf{v}})(\boldsymbol{\Sigma} \bar{\mathbf{v}})^T$$

Les auteurs ont confirmé et corrigé leur article.

Intégration des indices q-modulaires

☞ Étudions à présent un cas particulier des indices modulaires...

Intégration des indices q-modulaires

☞ Étudions à présent un cas particulier des indices modulaires...

$$\text{LWE} = \text{Système d'équations } q\text{-modulaires}$$

Indice q-modulaire « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{q}$ » permet de retirer une équation.

$$\Rightarrow \text{ Aussi « puissant » que les indices parfaits.}$$

Intégration des indices q-modulaires

Effet sur l'instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Nous connaissons $\mathbf{v} \in \mathbb{Z}^{m+n}$ et $l \in \mathbb{Z}$ tels que $\langle \mathbf{s}, \mathbf{v} \rangle \equiv l \pmod{q}$. Pour prendre en compte cet indice, nous devons transformer $\mathcal{I}$ en $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ avec

$$\begin{aligned}
\Lambda' &= \boldsymbol{\Pi} \cdot (\Lambda \cap \{\mathbf{x} \in \mathbb{Z}^{m+n+1} : \langle \mathbf{x}, \bar{\mathbf{v}} \rangle \equiv 0 \pmod{q}\}), \\
\boldsymbol{\mu}' &= \boldsymbol{\Pi} \cdot \boldsymbol{\mu}, \\
\boldsymbol{\Sigma}' &= \boldsymbol{\Pi} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Pi}^T,
\end{aligned}$$

où $\bar{\mathbf{v}} := (\mathbf{v}; -l)$ et $\boldsymbol{\Pi}$ est la matrice retirant la $p^{\text{ième}}$ coordonnée avec p un pivot possible de $\bar{\mathbf{v}}$ (*i.e.* $\bar{\mathbf{v}}_p \neq 0$).

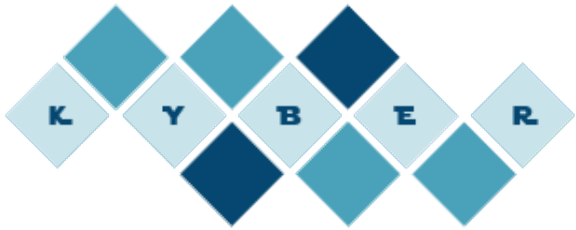
Table des matières

Les cryptosystèmes de Crystals

Mécanisme d'encapsulation de clé

Signature

Les cryptosystèmes de Crystals



Mécanisme
d'encapsulation de clé

Signature

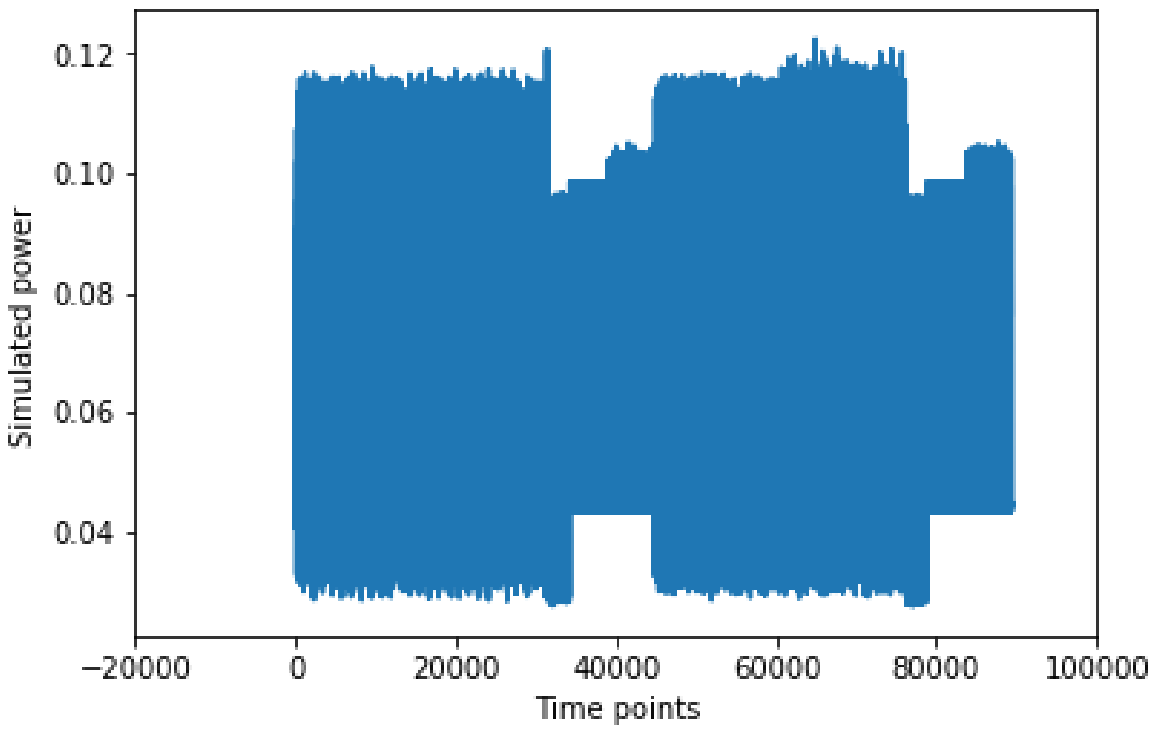
S'appuient sur un problème LWE structuré.

Utilisation de ELMO, un outil de simulation

```
--- 0x080001F0: 0xF835 bl 0x0800025B
--- 0x0800025C: 0x4B03 ldr r3,[PC+#0x800026C]
--- 0x0800025C: 0x4B03 ldr r3,[PC+#0x800026C]
--- 0x0800025E: 0x4A04 ldr r2,[PC+#0x8000270]
--- 0x0800025E: 0x4A04 ldr r2,[PC+#0x8000270]
--- 0x08000260: 0x4343 muls r3,r0
--- 0x08000262: 0x169B asrs r3,r3,#0x1A
--- 0x08000264: 0x4353 muls r3,r2
--- 0x08000266: 0x1AC0 subs r0,r0,r3
--- 0x08000268: 0xB200 sxth r0,r0
--- 0x0800026A: 0x4770 bx r14
--- 0x080001F2: 0x8020 strh r0,[r4,#0x200015E8]
--- 0x080001F2: 0x8020 strh r0,[r4,#0x200015E8]
```

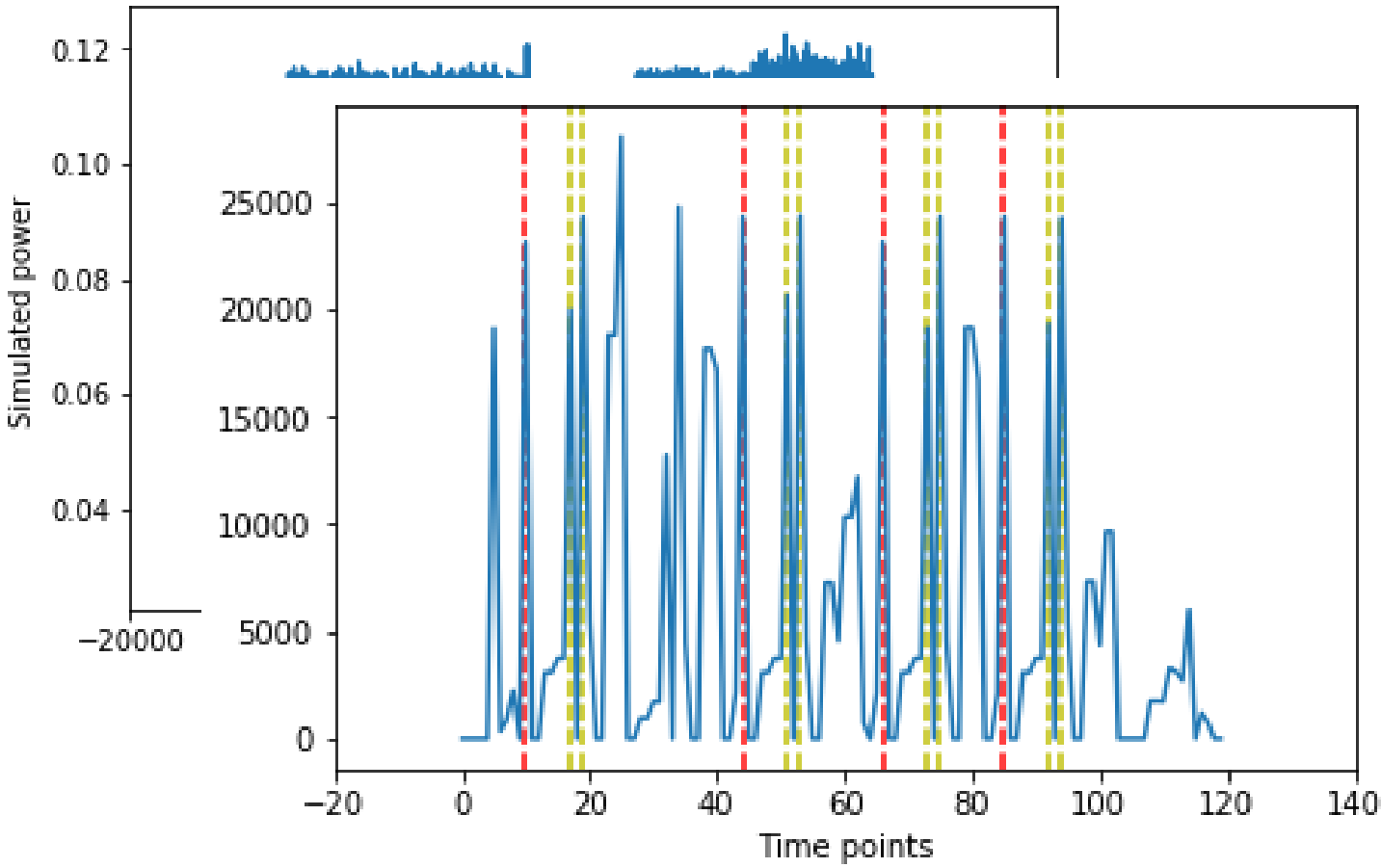
Utilisation de ELMO, un outil de simulation

```
--- 0x080001F0: 0xF835 bl 0x0800025B
--- 0x0800025C: 0x4B03 ldr r3,[PC+#0x800026C]
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
```



Utilisation de ELMO, un outil de simulation

```
--- 0x080001F0: 0xF835 bl 0x0800025B
--- 0x0800025C: 0x4B03 ldr r3,[PC+#0x800026C]
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
--- 0x08
```



Attaques par gabarits (*template attacks*)

Instruction assembleur étudiée : la **multiplication**.

En pratique, multiplication entre

- une variable connue $\omega \in \Omega$ et
- une variable inconnue dépendant du secret $a \in \Delta$.

Attaques par gabarits (*template attacks*)

Instruction assembleur étudiée : la **multiplication**.

En pratique, multiplication entre

- une variable connue $\omega \in \Omega$ et
- une variable inconnue dépendant du secret $a \in \Delta$.

Un « Gabarit » : la consommation **moyenne** de courant de la multiplication pour ω et a fixés

☞ Pré-déterminons les gabarits pour tous les a et ω possibles.

Attaques par gabarits (*template attacks*)

Instruction assembleur étudiée : la **multiplication**.

En pratique, multiplication entre

- une variable connue $\omega \in \Omega$ et
- une variable inconnue dépendant du secret $a \in \Delta$.

Un « Gabarit » : la consommation **moyenne** de courant de la multiplication pour ω et a fixés

☞ Pré-déterminons les gabarits pour tous les a et ω possibles.

L'attaque par canaux auxiliaires va nous fournir une consommation pour la multiplication pour un certain ω.
Par **maximum de vraisemblance**,
nous pourrons extrapoler la valeur de a.

Attaque contre Kyber

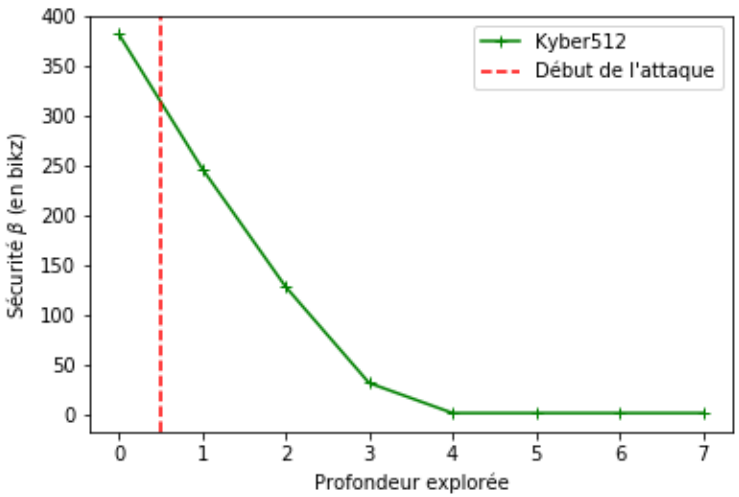
Mais que pouvons-nous attaquer ?

- Objectifs possibles :
  restauration de la clé privée ou du message ;
- Contextes possibles :
  génération de clé, chiffrement, déchiffrement ;
- Cibles possibles :
  multiplication matricielle, NTT, InvNTT, ...

Attaque contre Kyber

Attaque

- Contexte : la génération de clé
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0



Attaque contre Dilithium

Attaque

- Contexte : le processus de signature
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

Attaque contre Dilithium

Attaque

- Contexte : le processus de signature
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

Principe

Même attaque que celle contre la NTT de la génération de clé de Kyber :

- NTT(s) effectué à chaque signature, multi-traces possible ;
- $q = 8\ 380\ 417$, restriction aux deux premières itérations de la NTT.

Attaque contre Dilithium

Attaque

- Contexte : le processus de signature
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

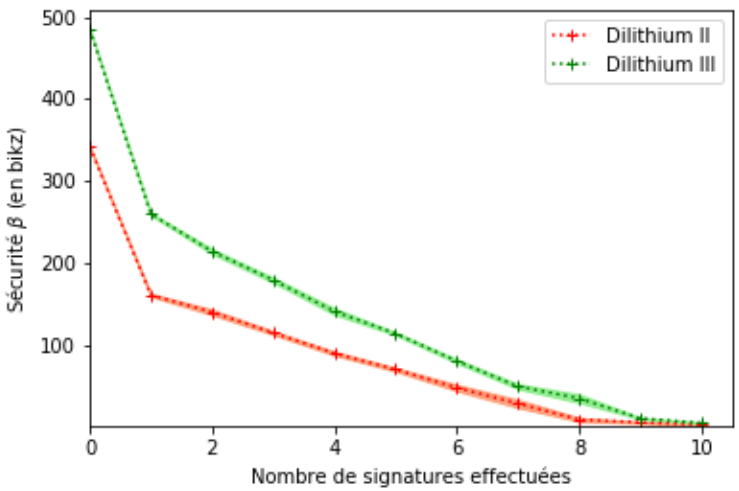


Table des matières

Conclusion

Les contributions du stage :

- Framework *Leaky-LWE-Estimator* plus complet et plus performant utilisable par un cryptanalyste ne maîtrisant pas la notion de réduction de réseaux ;
- Utilité du framework démontrée à travers diverses attaques réalistes.

Travaux prochainement disponibles sur
`https://github.com/lducas/leaky-LWE-Estimator`

Conclusion

Merci pour votre attention.

Avez-vous des questions ?

Le projet « *Post-Quantum Cryptography* » du NIST

Le Mercredi 22 Juillet 2020,

Passage au stade 3 du processus de standardisation

Les finalistes sont

Type	**PKE/KEM**	**Signature**
A base de réseaux	• Crystals-Kyber • NTRU • Saber	• Crystals-Dilithium • Falcon
A base de codes	Classic McEliece	
Multivarié		Rainbow

National Institute of Standards and Technology
U.S. Department of Commerce

Exemples de sécurité

Problème LWE : $(A \in \mathbb{Z}_q^{n \times m}, b = Az + e)$ avec $z \in \chi^m$ et $e \in \chi^n$.

	q	n	m	σ_χ	Sécurité	Sécurité
Frodo640	2^{15}	640	640	2.75	***484 bikz***	144-178 bits
Frodo1344	2^{16}	1344	1344	2.75	***930 bikz***	277-342 bits
Kyber512	3329	512	512	1.0	***380 bikz***	113-140 bits
Kyber1024	3329	1024	1024	1.0	***880 bikz***	262-324 bits
DilithiumII	8380417	1024	768	3.6	***340 bikz***	101-125 bits
DilithiumIV	8380417	1536	1280	1.1	***605 bikz***	180-222 bits

Indices utiles ?

Fuites par canaux auxiliaires :

- rarement linéaire par rapport à $\bar{\mathbf{s}}$;
- souvent une information binaire (ex. poids de Hamming).

Indices utiles ?

Fuites par canaux auxiliaires :

- rarement linéaire par rapport à $\bar{\mathbf{s}}$;
- souvent une information binaire (ex. poids de Hamming).

Leaky-LWE-Estimator prend en charge :

- Fuite directe d'un coefficient de $\bar{\mathbf{s}}$: $\bar{\mathbf{s}}_0 = l$;
- Fuite des bits de poids faible : $\bar{\mathbf{s}}_0 \equiv l \pmod 2$;

Indices utiles ?

Fuites par canaux auxiliaires :

- rarement linéaire par rapport à $\bar{\mathbf{s}}$;
- souvent une information binaire (ex. poids de Hamming).

Leaky-LWE-Estimator prend en charge :

- Fuite directe d'un coefficient de $\bar{\mathbf{s}}$: $\bar{\mathbf{s}}_0 = l$;
- Fuite des bits de poids faible : $\bar{\mathbf{s}}_0 \equiv l \pmod 2$;
- Informations linéaires d'une fuite non-linéaire : $H(\bar{\mathbf{s}}_0) = 2$
  Alors si $\mathrm{supp}(s_i) = \{-5, ..., 5\}$, alors $\bar{\mathbf{s}}_0 \in \{3, 5\}$.
  - ☞ Un indice modulaire : $\bar{\mathbf{s}}_0 \equiv 2 \pmod 2$
  - ☞ Un indice approximé : $\bar{\mathbf{s}}_0 = 4 + \varepsilon$ avec $\sigma_\varepsilon^2 = 1$

Réduction de dimension

Réduction de dimension

Soit une instance DBDD $\mathcal{I} = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Il est possible de transformer $\mathcal{I}$ en une instance $\mathcal{I}' = (\Lambda', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$ dont le réseau est de rang maximal.

Si nous avons $\mathbf{\Gamma}$ tel que $\Lambda = \mathbf{\Gamma}\Lambda'$ avec Λ' de rang maximal et $\text{rank}(\Lambda) = \text{rank}(\Lambda')$, alors

$$\boldsymbol{\mu}' \leftarrow (\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T \cdot \boldsymbol{\mu},$$
$$\boldsymbol{\Sigma}' \leftarrow (\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T \cdot \boldsymbol{\Sigma} \cdot \mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1},$$

Réduction de dimension

Après l'intégration d'un indice parfait « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ », le $\mathbf{\Gamma}$ suivant vérifie les hypothèses du théorème.

$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{v}_1^T & -\mathbf{v}_2^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

où $\mathbf{v}_1$ et $\mathbf{v}_2$ sont définis par la relation $\bar{\mathbf{v}}^T = (\mathbf{v}_1^T, 1, \mathbf{v}_2^T)$.

Réduction de dimension

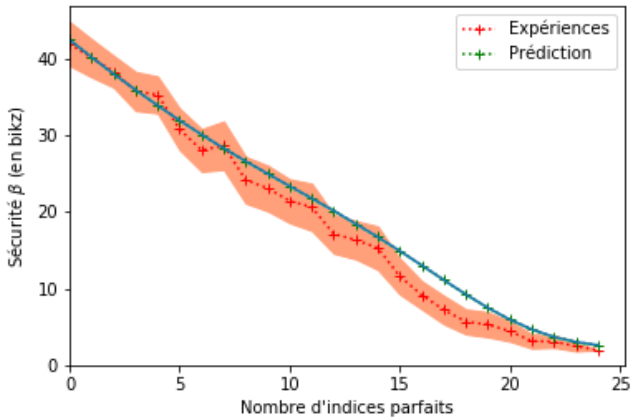
Après l'intégration d'un indice parfait « $\langle \bar{\mathbf{s}}, \bar{\mathbf{v}} \rangle = 0$ », le $\mathbf{\Gamma}$ suivant vérifie les hypothèses du théorème.

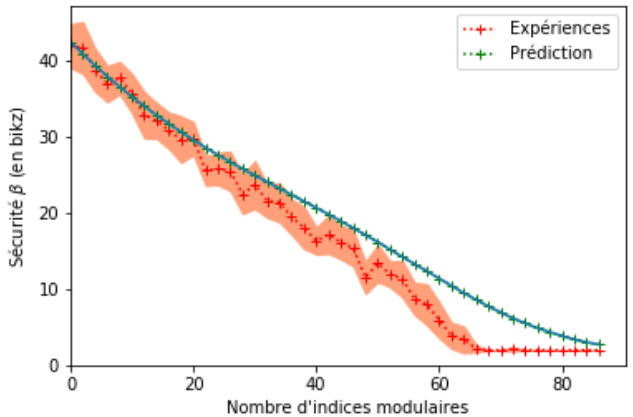
$$\mathbf{\Gamma} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{v}_1^T & -\mathbf{v}_2^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

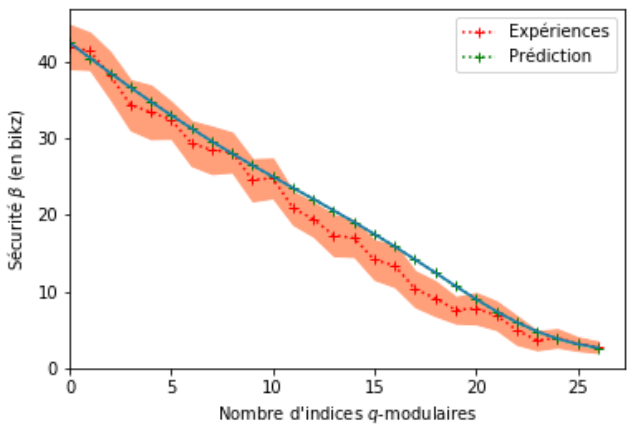
où $\mathbf{v}_1$ et $\mathbf{v}_2$ sont définis par la relation $\bar{\mathbf{v}}^T = (\mathbf{v}_1^T, 1, \mathbf{v}_2^T)$.

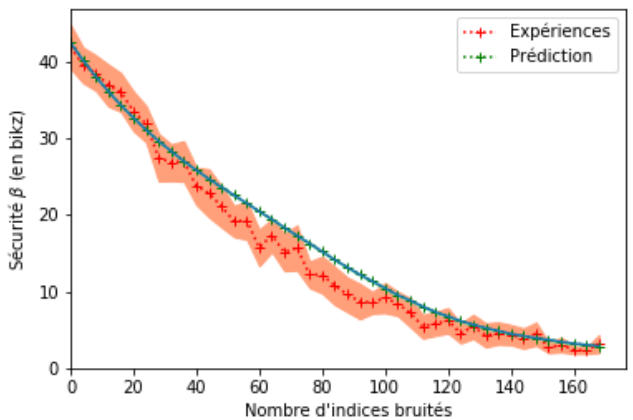
☞ À présent, **les instances DBDD manipulées ont toutes leur réseau de rang maximal**.

Validations des améliorations









ELMO, un outil de simulation

Qu'est-ce que ELMO ?

Outil permettant de simuler la consommation de courant d'un programme exécuté sur un processeur ARM Cortex M0.

Le modèle de fuite

$$y = \delta + [\mathbf{I_p}|\mathbf{I_s}|\mathbf{D}|\mathbf{DxIp}|\mathbf{DxIs}]\boldsymbol{\beta} + \varepsilon$$

Qualitativement

La consommation d'énergie d'une instruction dépend :

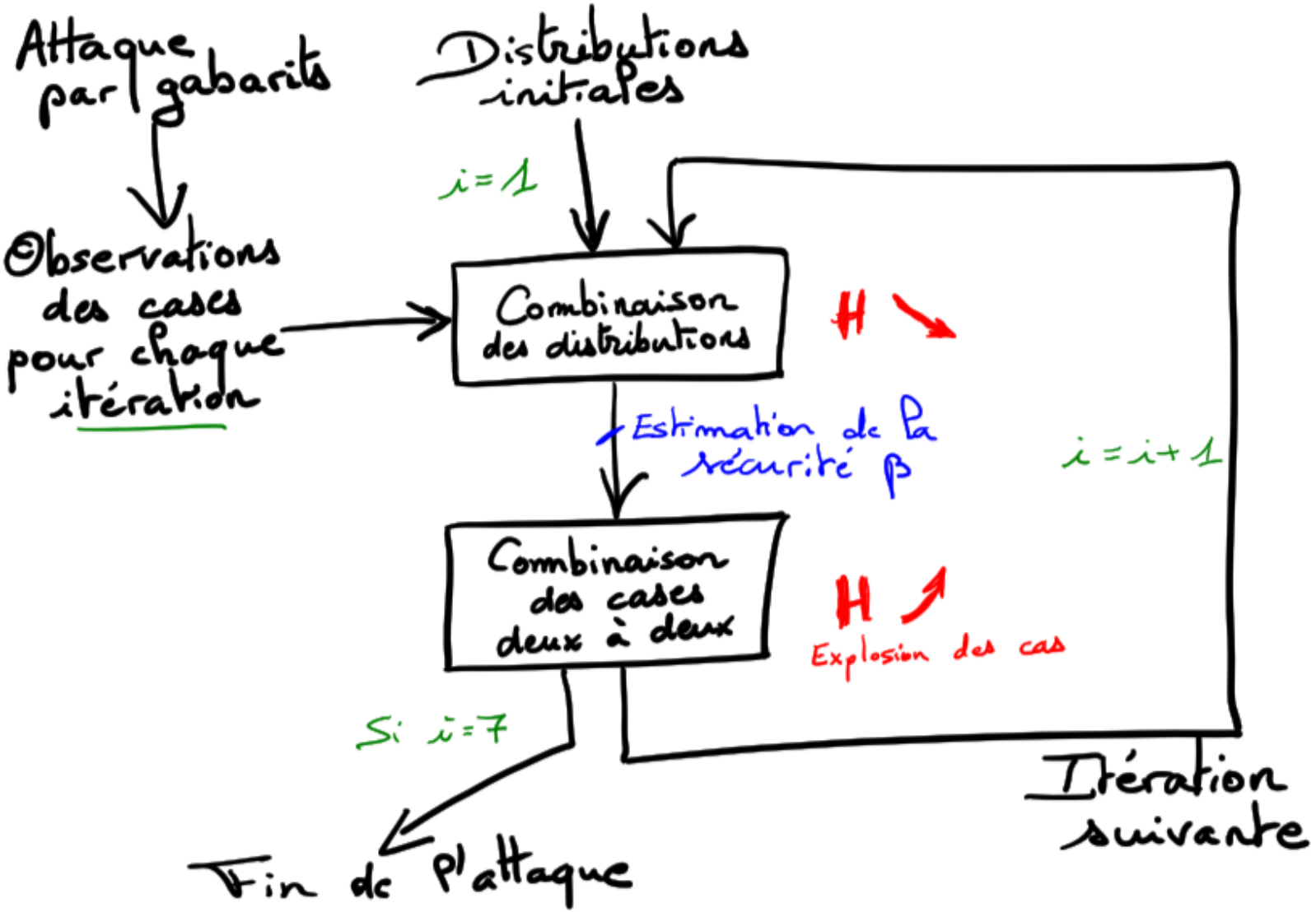
- du type de l'instruction précédente et de ses deux opérandes ;
- du type de l'instruction courante et de ses deux opérandes ;
- du type de l'instruction suivante.

Attaque principale contre Kyber

Attaque

- Contexte : la génération de clé
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

Attaque principale contre Kyber



Attaque principale contre Kyber

Attaque 1

- Contexte : la génération de clé
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

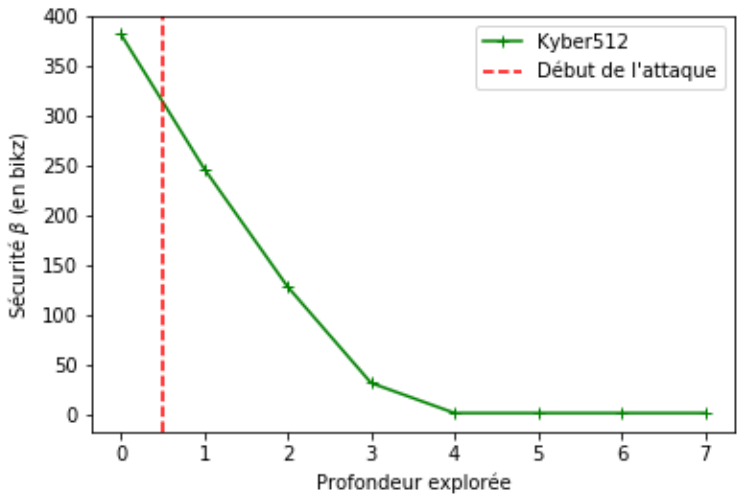
Principe

L'attaque consiste donc à trouver un juste milieu entre la divergence provoquée par la combinaison des cases deux à deux et une élimination modérée des possibilités.

Attaque principale contre Kyber

Attaque 1

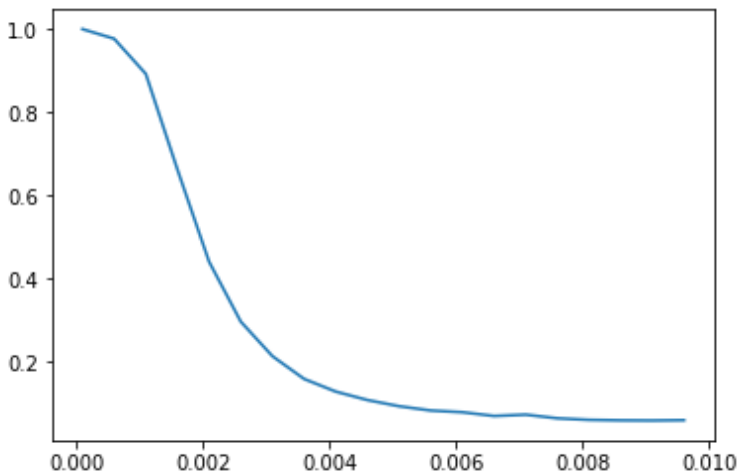
- Contexte : la génération de clé
- Cible : la NTT (*Number Theoretic Transform*)
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0



Autre attaque contre Kyber

Attaque contre Kyber

- Contexte : la génération de clé
- Cible : la multiplication matricielle points par points
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0

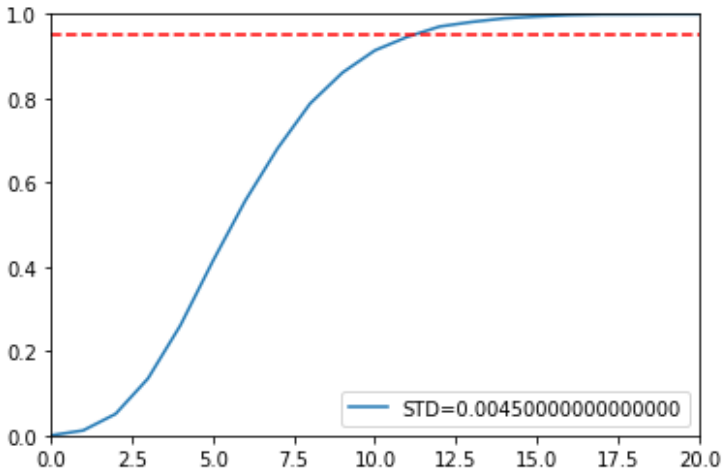


Taux de restauration
de la NTT du secret

Autre attaque contre Kyber

Attaque contre Kyber

- Contexte : le déchiffrement
- Cible : la multiplication matricielle points par points
- Instruction assembleur attaquée : la multiplication
- Composant attaqué : Processeur ARM Cortex M0



Taux de restauration de la NTT du secret en fonction du nombre de déchiffrements.