

CRYPTO 2025

ePrint: 2025/575  
arXiv: 2503.23238

# Wagner's algorithm provably runs in subexponential time for $\text{SIS}^\infty$

*Léo Ducas, Lynn Engelberts, Johanna Loyer*



Research institute for mathematics &  
computer science in the Netherlands



# Motivation

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  **$n, m, q$**

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  $n, m, q$

Kirchner and Fouque (2015): A claimed **subexponential-time** algorithm for LWE with **narrow** error distribution

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  $n, m, q$

Kirchner and Fouque (2015): A claimed **subexponential-time** algorithm for LWE with **narrow** error distribution

❖ *Original statement:* Holds when  $m \approx n$  and  $q = n^{\Theta(1)}$

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  **$n, m, q$**

Kirchner and Fouque (2015): A claimed **subexponential-time** algorithm for LWE with **narrow** error distribution

- ❖ *Original statement:* Holds when  **$m \approx n$**  and  $q = n^{\Theta(1)}$
- ❖ Herold, Kirshanova, and May (2018) found an error in the proof, which they fixed for  **$m \approx n \ln n$**



# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  $n, m, q$

Kirchner and Fouque (2015): A claimed **subexponential-time** algorithm for LWE with **narrow** error distribution

- ❖ *Original statement:* Holds when  $m \approx n$  and  $q = n^{\Theta(1)}$
- ❖ Herold, Kirshanova, and May (2018) found an error in the proof, which they fixed for  $m \approx n \ln n$
- ❖ *Common belief:* It can also be resolved for  $m \approx n$

# Motivation

Learning with Errors (**LWE**) is one of the most prominent computational problems in post-quantum cryptography

*Task:* Solve a ‘noisy’ system of  $m$  linear equations in  $n$  variables modulo  $q$

 *Each equation is perturbed by some **random** error*

Hardness of LWE depends on the parameters  $n, m, q$

Kirchner and Fouque (2015): A claimed **subexponential-time** algorithm for LWE with **narrow** error distribution

- ❖ *Original statement:* Holds when  $m \approx n$  and  $q = n^{\Theta(1)}$
- ❖ Herold, Kirshanova, and May (2018) found an error in the proof, which they fixed for  $m \approx n \ln n$
- ❖ *Common belief:* It can also be resolved for  $m \approx n$

Our work: We establish this for the **first step** of Kirchner and Fouque’s algorithm

# Contribution

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ , **SIS** <sub>$n, m, q, \beta$</sub>  <sup>$\infty$</sup>  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ , **SIS** <sub>$n, m, q, \beta$</sub>  <sup>$\infty$</sup>  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\mathbf{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$



# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ , **SIS** <sub>$n, m, q, \beta$</sub>  <sup>$\infty$</sup>  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$
- $\|\mathbf{x}\|_\infty \leq \beta$

# Contribution

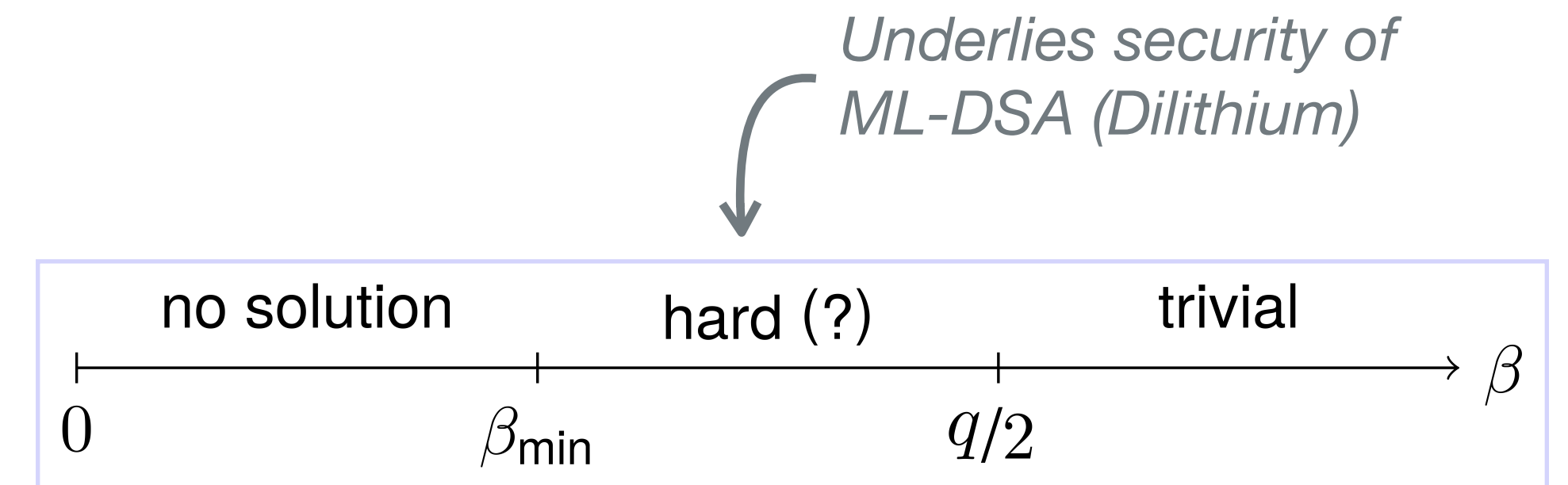
Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$
- $\|\mathbf{x}\|_\infty \leq \beta$



# Contribution

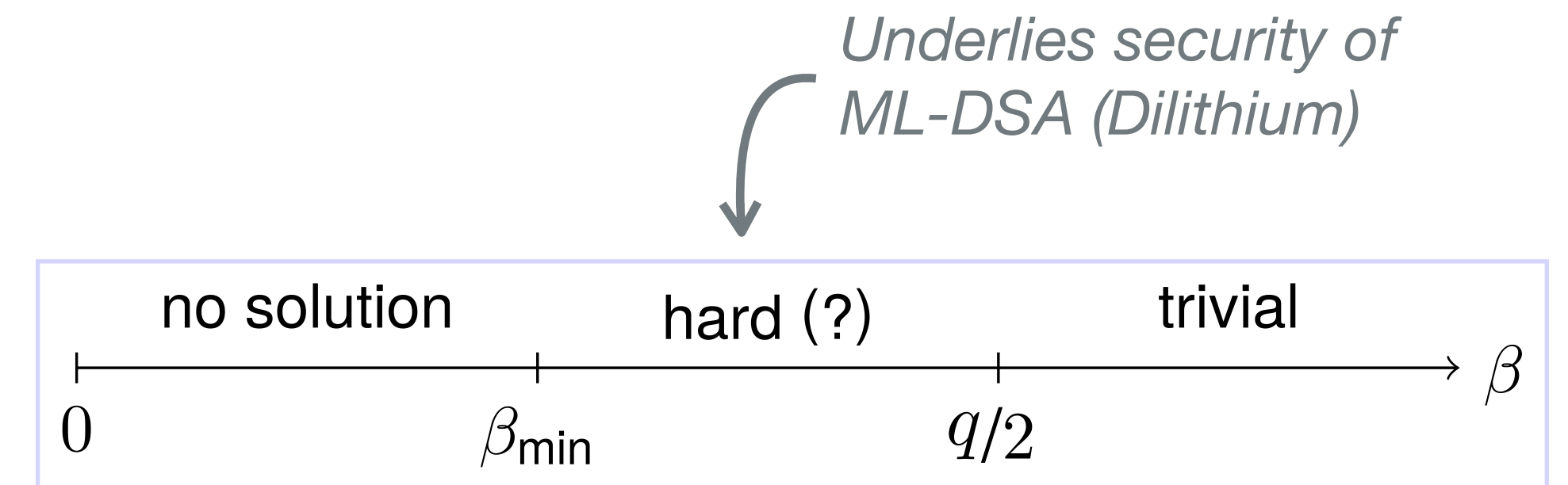
Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$
- $\|\mathbf{x}\|_\infty \leq \beta$



Our contribution:

# Contribution

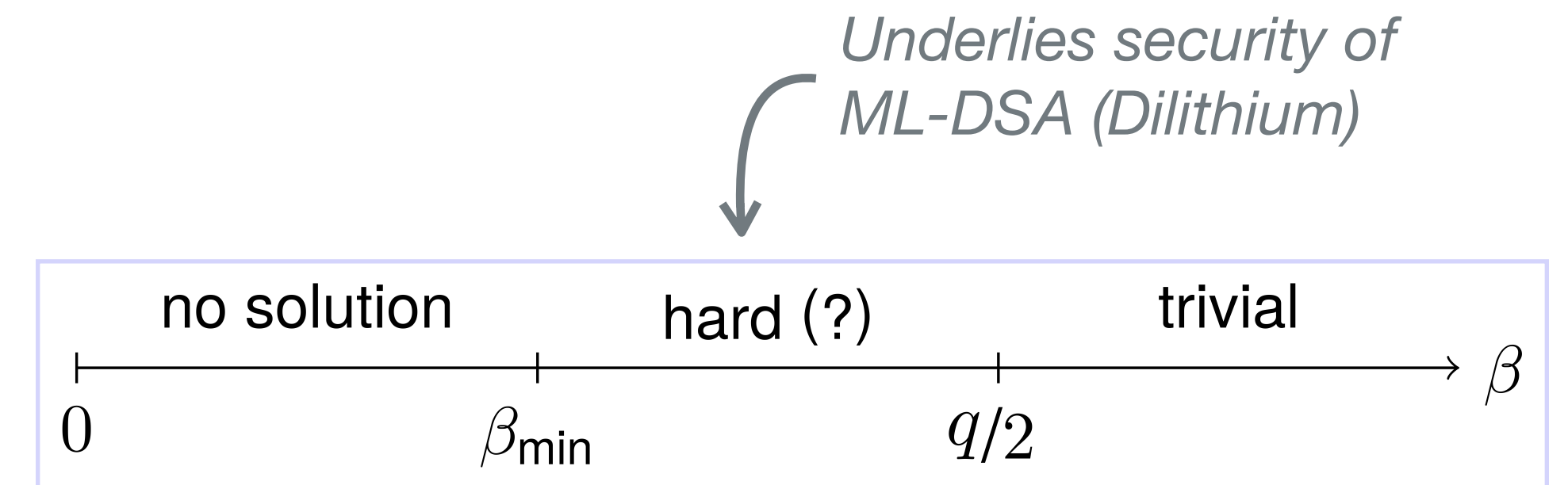
Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$
- $\|\mathbf{x}\|_\infty \leq \beta$



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

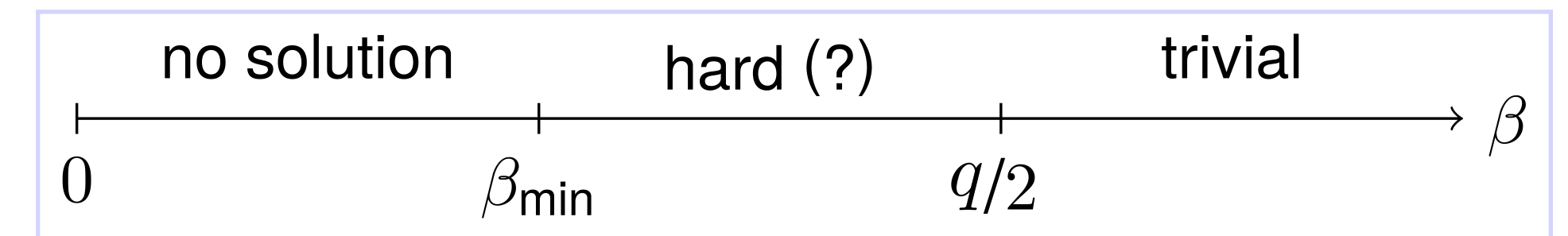
For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \pmod{q}$
- $\|\mathbf{x}\|_\infty \leq \beta$

*Underlies security of  
ML-DSA (Dilithium)*



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**
- ❖ Subexponential-time  $\text{SIS}_{n,m,q,\beta}^\infty$  algorithm for all  $\underbrace{m = n + \omega(n/\ln \ln n)}_{\text{Instead of } m \approx n \ln n}$  with  $m = n^{O(1)}$ , all prime  $q = n^{\Theta(1)}$ , and some **nontrivial  $\beta$**   
*Such as  $\beta = \frac{q}{\text{polylog}(n)}$*

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

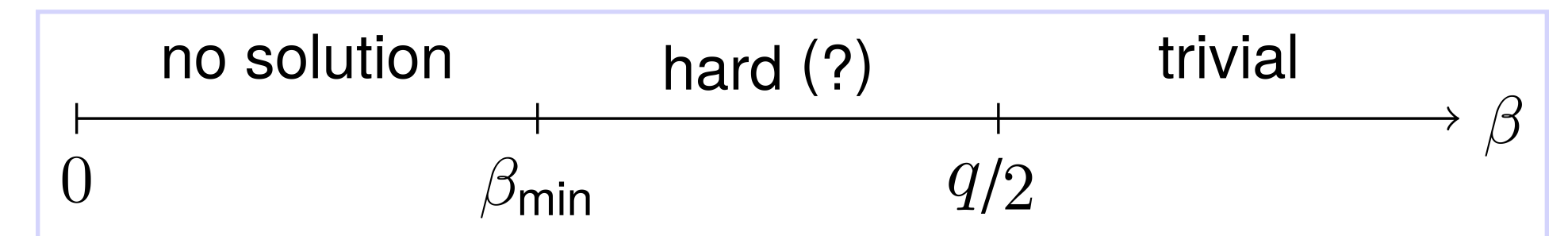
For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \pmod{q}$
- $\|\mathbf{x}\|_\infty \leq \beta$

*Underlies security of  
ML-DSA (Dilithium)*



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**
- ❖ Subexponential-time  $\text{SIS}_{n,m,q,\beta}^\infty$  algorithm for all  $\underbrace{m = n + \omega(n/\ln \ln n)}_{\text{Instead of } m \approx n \ln n}$  with  $m = n^{O(1)}$ , all prime  $q = n^{\Theta(1)}$ , and some **nontrivial  $\beta$**   
*Such as  $\beta = \frac{q}{\text{polylog}(n)}$*

Implications:

# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

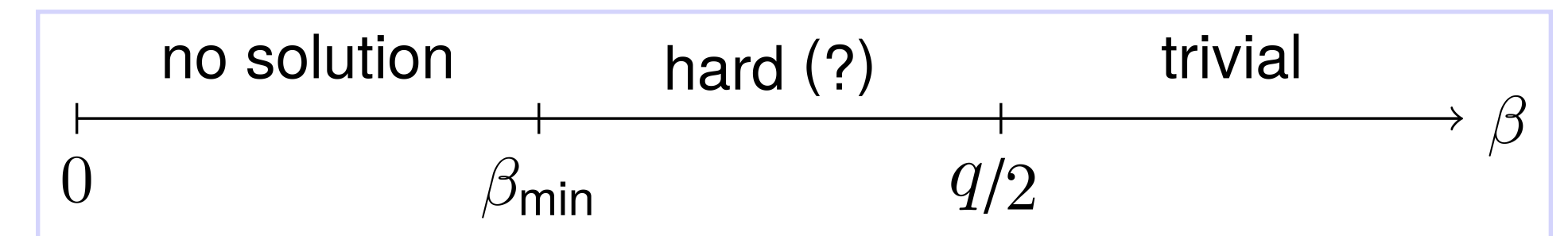
For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \pmod{q}$
- $\|\mathbf{x}\|_\infty \leq \beta$

*Underlies security of  
ML-DSA (Dilithium)*



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**
- ❖ Subexponential-time  $\text{SIS}_{n,m,q,\beta}^\infty$  algorithm for all  $\underbrace{m = n + \omega(n/\ln \ln n)}_{\text{Instead of } m \approx n \ln n}$  with  $m = n^{O(1)}$ , all prime  $q = n^{\Theta(1)}$ , and some **nontrivial  $\beta$**   
Such as  $\beta = \frac{q}{\text{polylog}(n)}$

Implications:

- ❖ Result does **not** affect the **concrete security of ML-DSA**



# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), derived from the  **$m$**  LWE equations

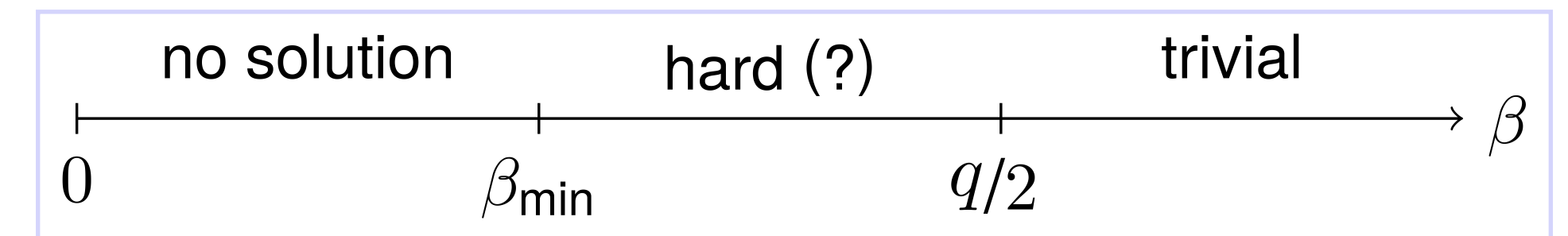
For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  **$\text{SIS}_{n,m,q,\beta}^\infty$**  is defined as:

**Given:** Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

**Goal:** Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \bmod q$
- $\|\mathbf{x}\|_\infty \leq \beta$

Underlies security of  
ML-DSA (Dilithium)



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**
- ❖ Subexponential-time  $\text{SIS}_{n,m,q,\beta}^\infty$  algorithm for all  **$m = n + \omega(n/\ln \ln n)$**  with  $m = n^{O(1)}$ , all prime  $q = n^{\Theta(1)}$ , and some **nontrivial  $\beta$**   
*Instead of  $m \approx n \ln n$*  *Such as  $\beta = \frac{q}{\text{polylog}(n)}$*

Implications:

- ❖ Result does **not** affect the **concrete security of ML-DSA**
- ❖ Subexponential time also applies to nontrivial instances of **Inhomogeneous-SIS** and  **$\text{SIS}^\times$** , variants of SIS in the  $\ell_2$ -norm



# Contribution

Kirchner and Fouque's first step solves an instance of the **Short Integer Solution** problem (SIS), *derived from the  $m$  LWE equations*

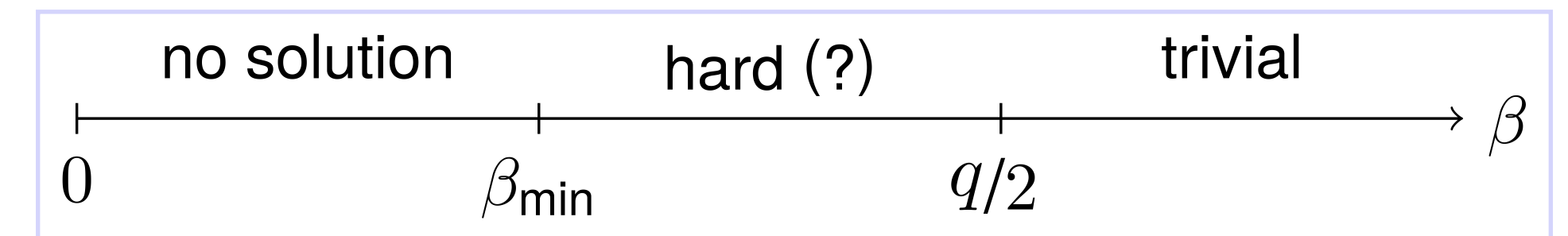
For  $n, m, q \in \mathbb{N}$  and norm bound  $\beta > 0$ ,  $\text{SIS}_{n,m,q,\beta}^\infty$  is defined as:

*Given:* Uniformly random matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

*Goal:* Find nonzero  $\mathbf{x} \in \mathbb{Z}^m$  satisfying:

- $\mathbf{Ax} = \mathbf{0} \pmod{q}$
- $\|\mathbf{x}\|_\infty \leq \beta$

*Underlies security of  
ML-DSA (Dilithium)*



Our contribution:

- ❖ **Re-interpretation** of the Kirchner-Fouque SIS step as going through a chain of projected **lattices**
- ❖ Subexponential-time  $\text{SIS}_{n,m,q,\beta}^\infty$  algorithm for all  $\underbrace{m = n + \omega(n/\ln \ln n)}_{\text{Instead of } m \approx n \ln n}$  with  $m = n^{O(1)}$ , all prime  $q = n^{\Theta(1)}$ , and some **nontrivial  $\beta$**   
*Such as  $\beta = \frac{q}{\text{polylog}(n)}$*

Implications:

- ❖ Result does **not** affect the **concrete security of ML-DSA**
- ❖ Subexponential time also applies to nontrivial instances of **Inhomogeneous-SIS** and **SIS<sup>x</sup>**, variants of SIS in the  $\ell_2$ -norm
- ❖ No direct corollary yet for LWE (for technical reasons), but seems feasible

# Wagner-style algorithm for $\text{SIS}^\infty$

# Wagner-style algorithm for $\text{SIS}^\infty$

Kirchner-Fouque SIS step: Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

# Wagner-style algorithm for $\text{SIS}^\infty$

**Kirchner-Fouque SIS step:** Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

# Wagner-style algorithm for $\text{SIS}^\infty$

**Kirchner-Fouque SIS step:** Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$

# Wagner-style algorithm for $\text{SIS}^\infty$

**Kirchner-Fouque SIS step:** Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \bmod q$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

# Wagner-style algorithm for $\text{SIS}^\infty$

**Kirchner-Fouque SIS step:** Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod{q}$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod{q}$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

**Strategy:**

# Wagner-style algorithm for $\text{SIS}^\infty$

**Kirchner-Fouque SIS step:** Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

**Strategy:**

1. Sample many  $\mathbf{z} \in \mathbb{Z}^{m-n}$  with  $\|\mathbf{z}\|_\infty$  small



# Wagner-style algorithm for $\text{SIS}^\infty$

Kirchner-Fouque SIS step: Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

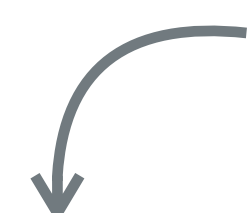
Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

**Strategy:**

1. Sample many  $\mathbf{z} \in \mathbb{Z}^{m-n}$  with  $\|\mathbf{z}\|_\infty$  small

*Note that  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  may not be small*



# Wagner-style algorithm for $\text{SIS}^\infty$

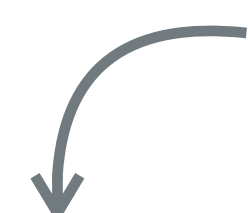
Kirchner-Fouque SIS step: Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

**Strategy:**

*Note that  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  may not be small*



1. Sample many  $\mathbf{z} \in \mathbb{Z}^{m-n}$  with  $\|\mathbf{z}\|_\infty$  small
2. Iteratively **combine** vectors to obtain solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  that satisfy  $\|\mathbf{x}\|_\infty \leq \beta$

# Wagner-style algorithm for $\text{SIS}^\infty$

Kirchner-Fouque SIS step: Refined version of **Wagner's algorithm** (originally solving the generalized birthday problem)

Let  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  be an  $\text{SIS}_{n,m,q,\beta}^\infty$  instance:

- ❖ Assuming  $\mathbf{A}$  is of full rank, we can write  $\mathbf{A} = [\mathbf{A}' \mid \mathbf{I}_n]$  for  $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$
- ❖ All solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  are of the form  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  for some  $\mathbf{z} \in \mathbb{Z}^{m-n}$

**Strategy:**

1. Sample many  $\mathbf{z} \in \mathbb{Z}^{m-n}$  with  $\|\mathbf{z}\|_\infty$  small
2. Iteratively **combine** vectors to obtain solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$  that satisfy  $\|\mathbf{x}\|_\infty \leq \beta$

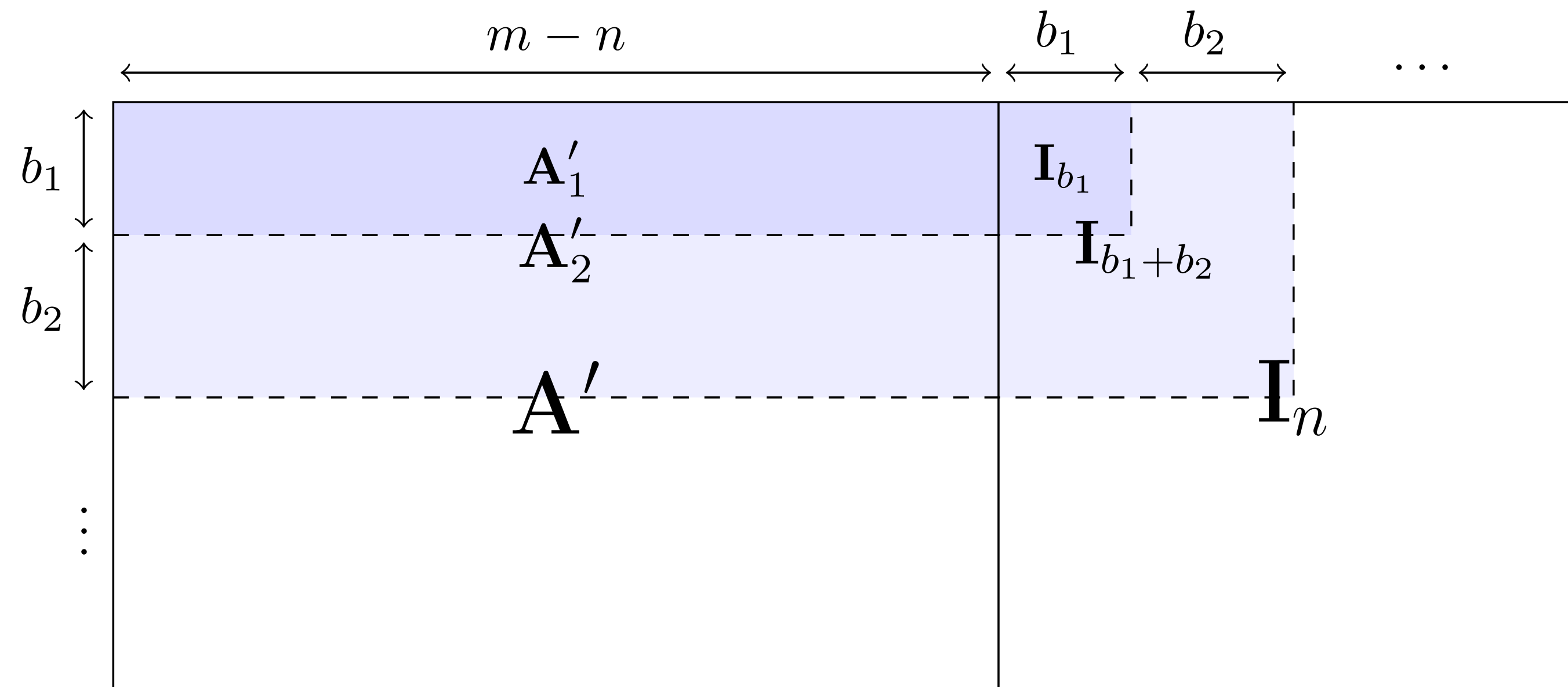
*Note that  $\mathbf{x} = \begin{pmatrix} \mathbf{z} \\ -\mathbf{A}'\mathbf{z} \end{pmatrix} \pmod q$  may not be small*

*Wagner-style: Considering one block of coordinates at a time*

# Wagner-style algorithm for $\text{SIS}^\infty$

# Wagner-style algorithm for $\text{SIS}^\infty$

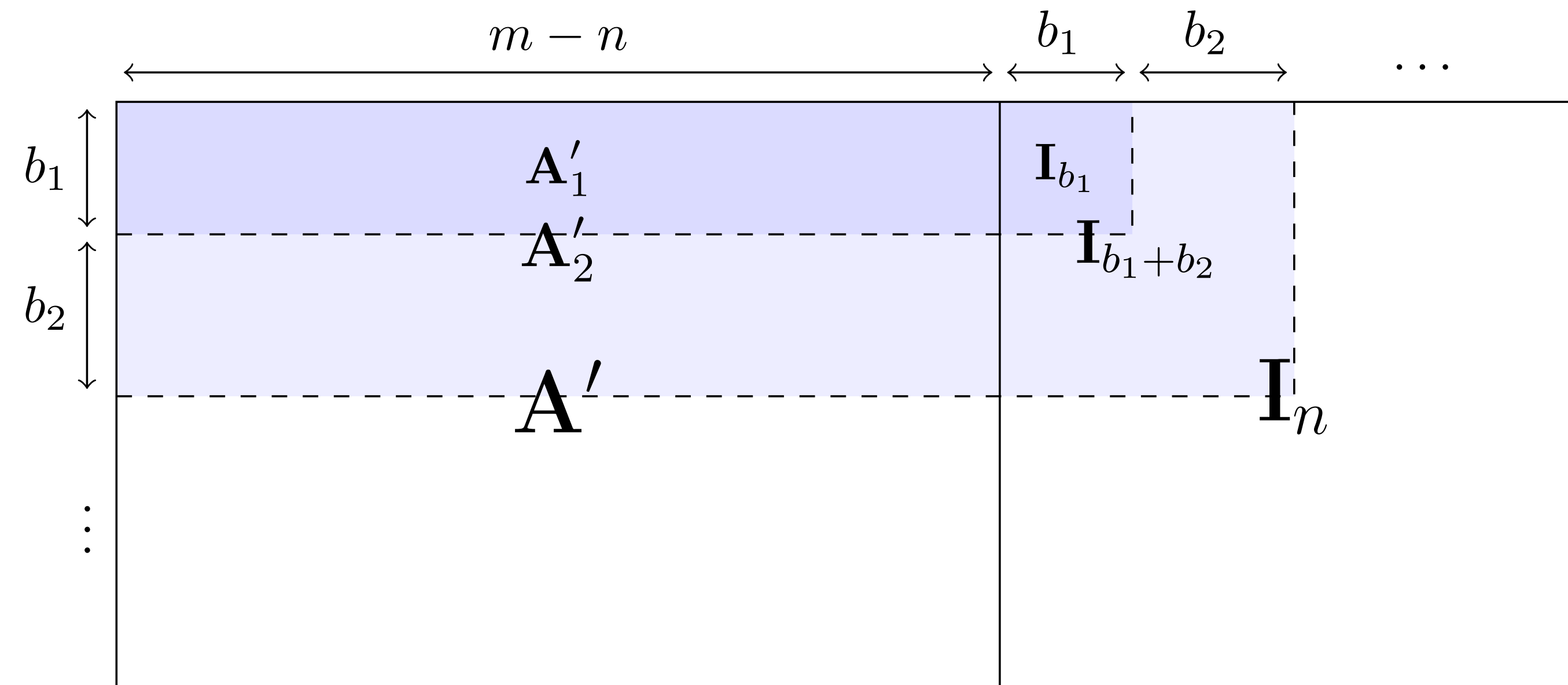
Consider  $b_1, \dots, b_r \in \mathbb{N}$  with  $\sum_{i=1}^r b_i = n$



# Wagner-style algorithm for $\text{SIS}^\infty$

Consider  $b_1, \dots, b_r \in \mathbb{N}$  with  $\sum_{i=1}^r b_i = n$

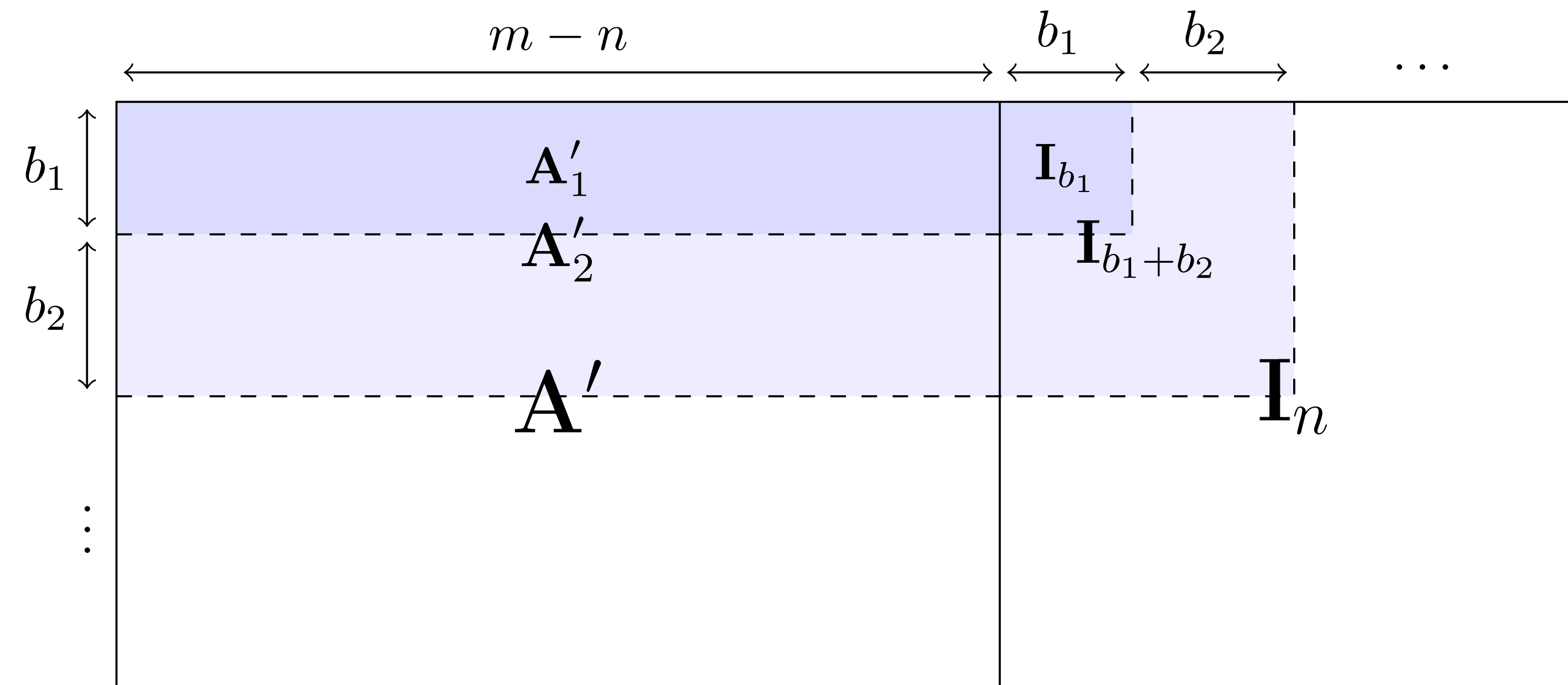
This yields matrices  $\mathbf{A}_1, \dots, \mathbf{A}_r$  defined by  $\mathbf{A}_i := [\mathbf{A}'_i \mid \mathbf{I}_{b_1+\dots+b_i}]$ , where  $\mathbf{A}'_i$  consists of the first  $b_1 + \dots + b_i$  rows of  $\mathbf{A}'$



# Wagner-style algorithm for $\text{SIS}^\infty$

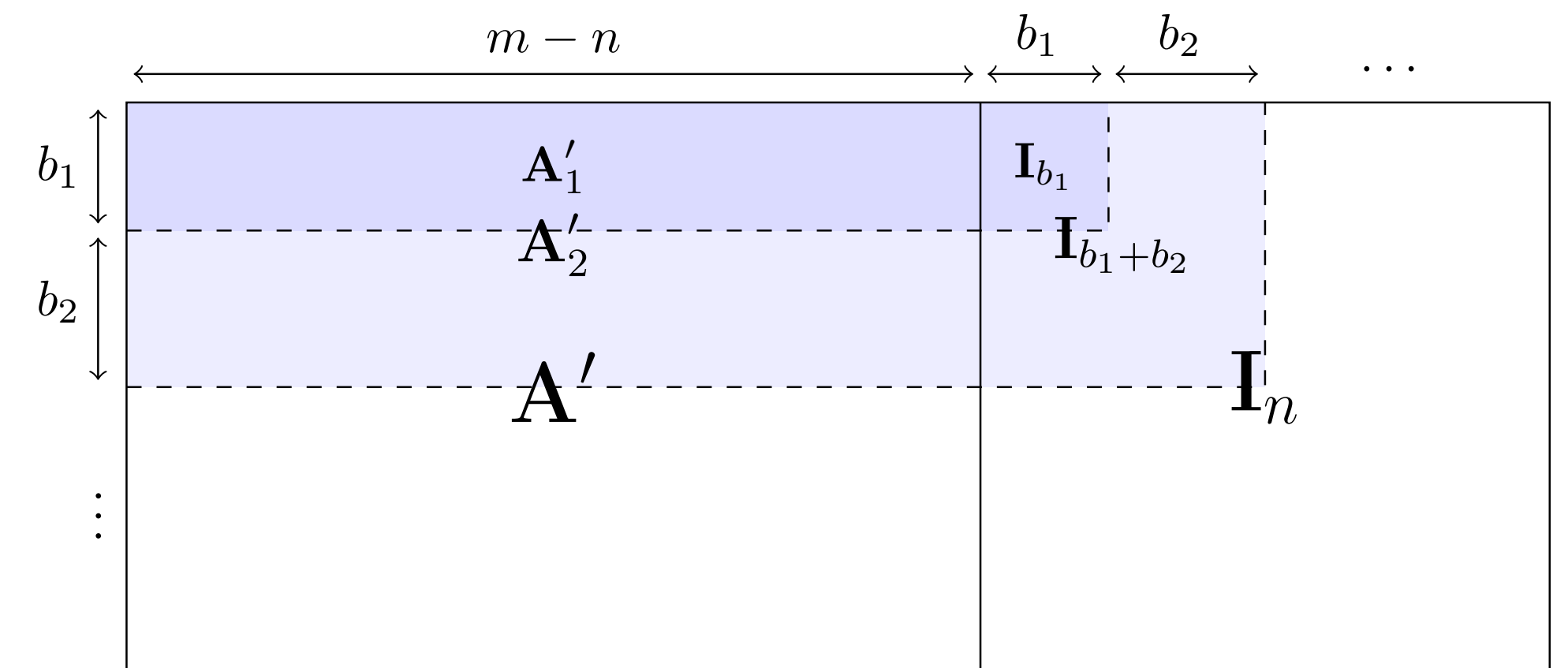
Consider  $b_1, \dots, b_r \in \mathbb{N}$  with  $\sum_{i=1}^r b_i = n$

This yields matrices  $\mathbf{A}_1, \dots, \mathbf{A}_r$  defined by  $\mathbf{A}_i := [\mathbf{A}'_i \mid \mathbf{I}_{b_1+\dots+b_i}]$ , where  $\mathbf{A}'_i$  consists of the first  $b_1 + \dots + b_i$  rows of  $\mathbf{A}'$



Iteratively, the algorithm constructs *short* solutions to  $\mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q$  from *short* solutions to  $\mathbf{A}_{i-1} \mathbf{x} = \mathbf{0} \bmod q$

# Kirchner and Fouque's algorithm for $\text{SIS}^\infty$

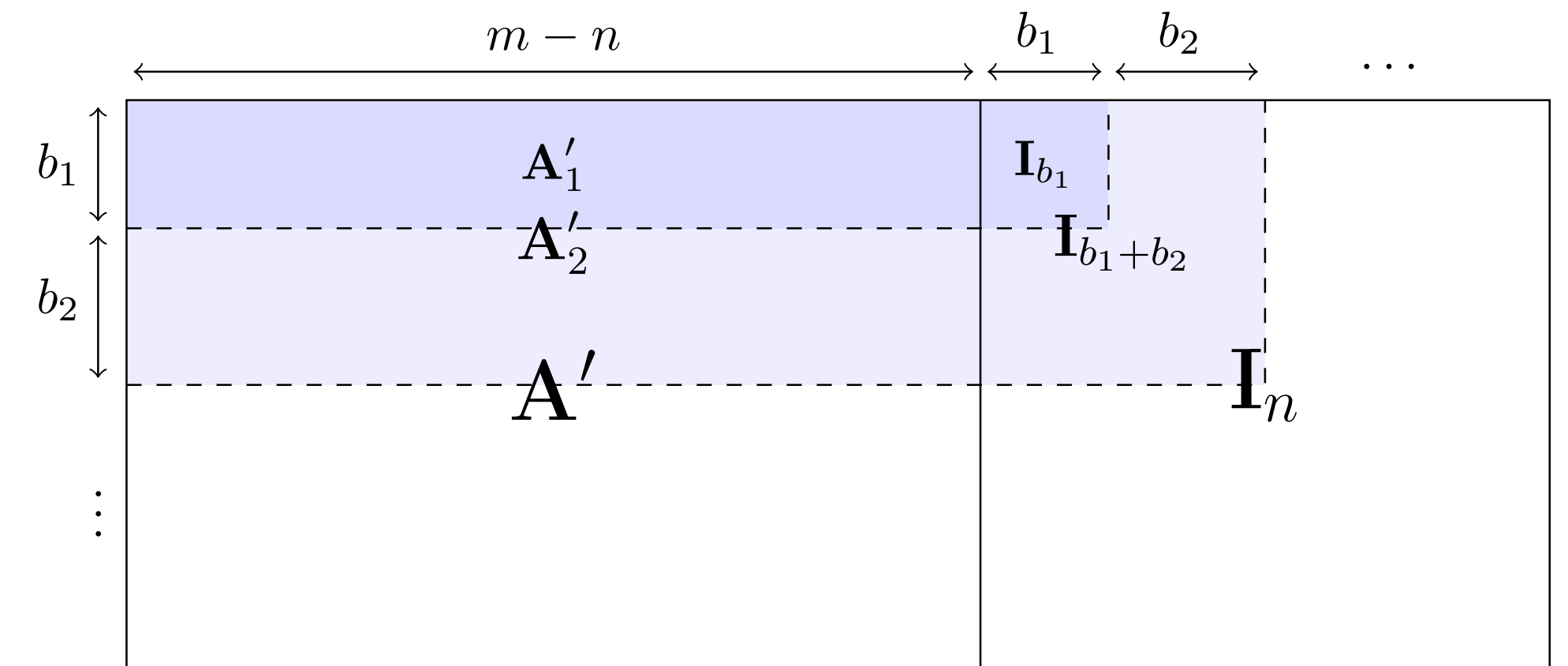




# Kirchner and Fouque's algorithm for $\text{SIS}^\infty$

**Algorithm:**

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

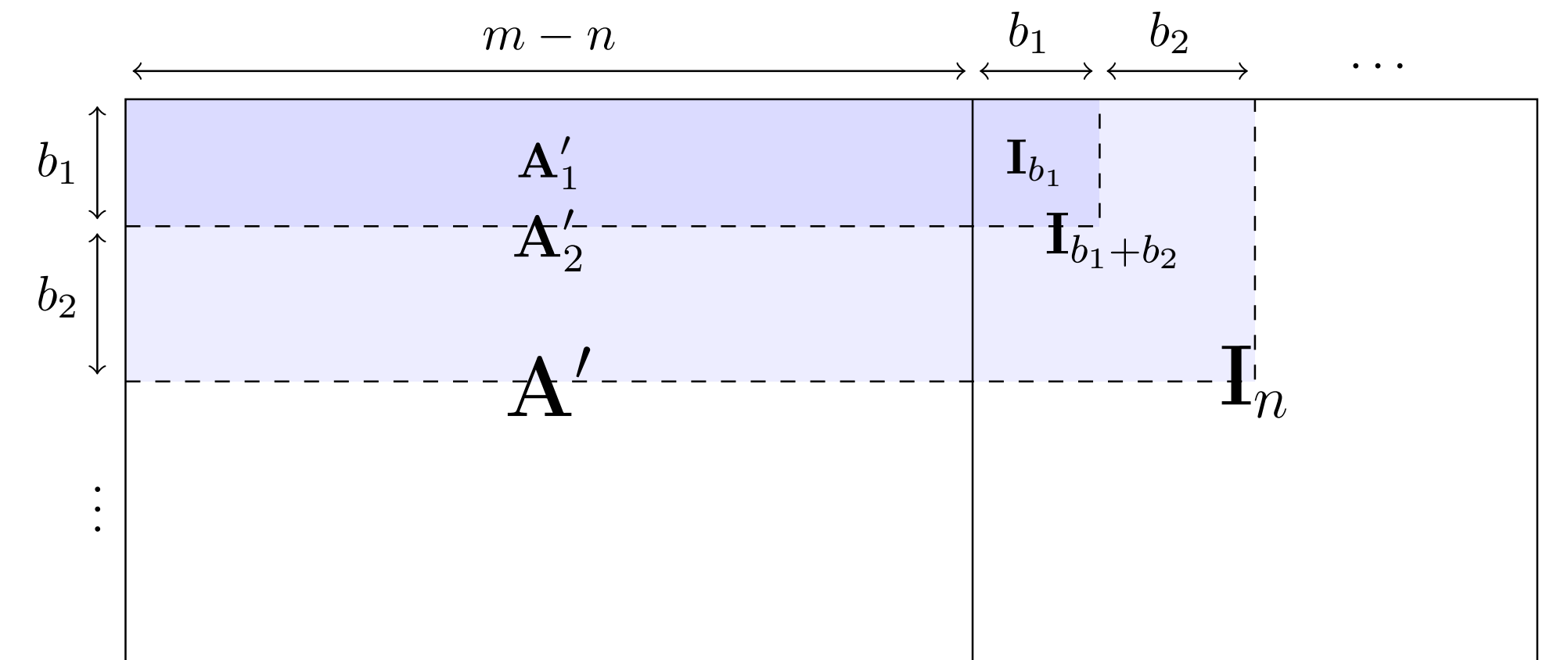


# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

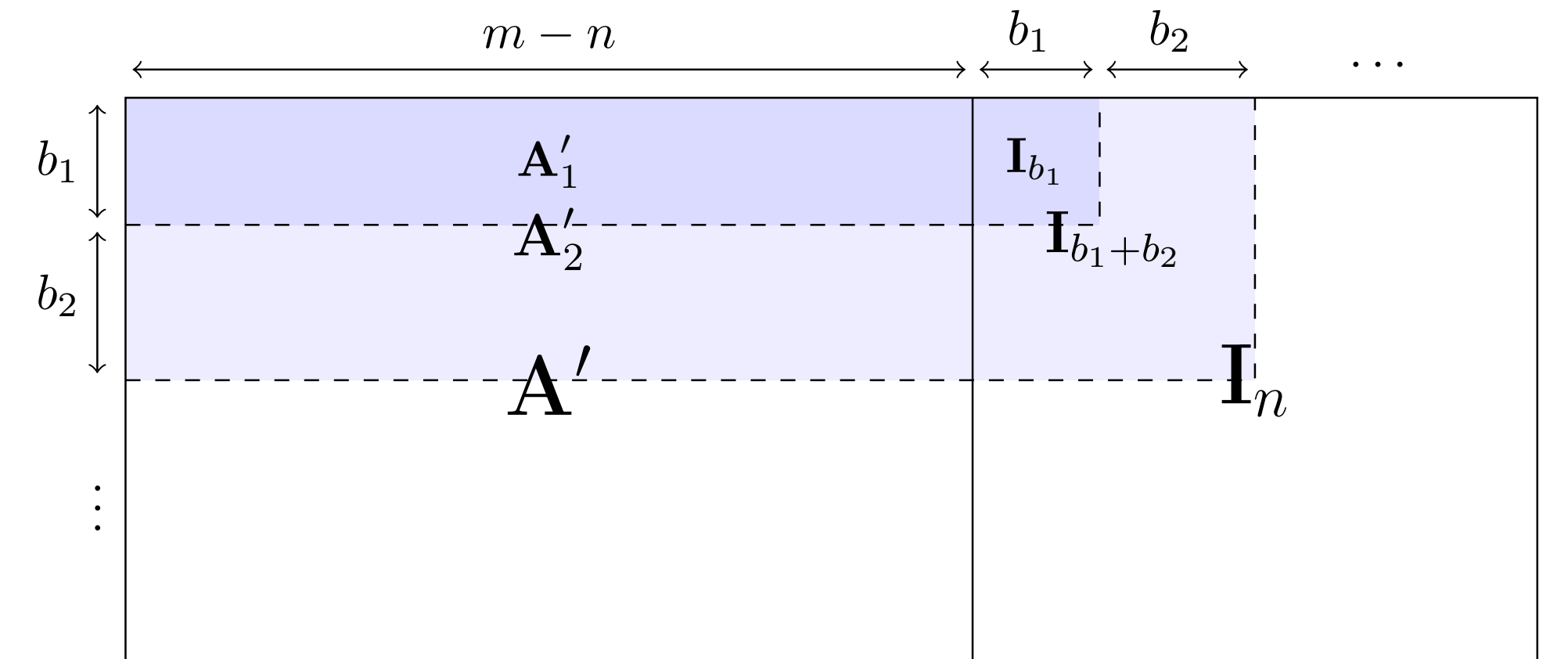


# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$
2. For  $i = 1, \dots, r$ :
  - I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

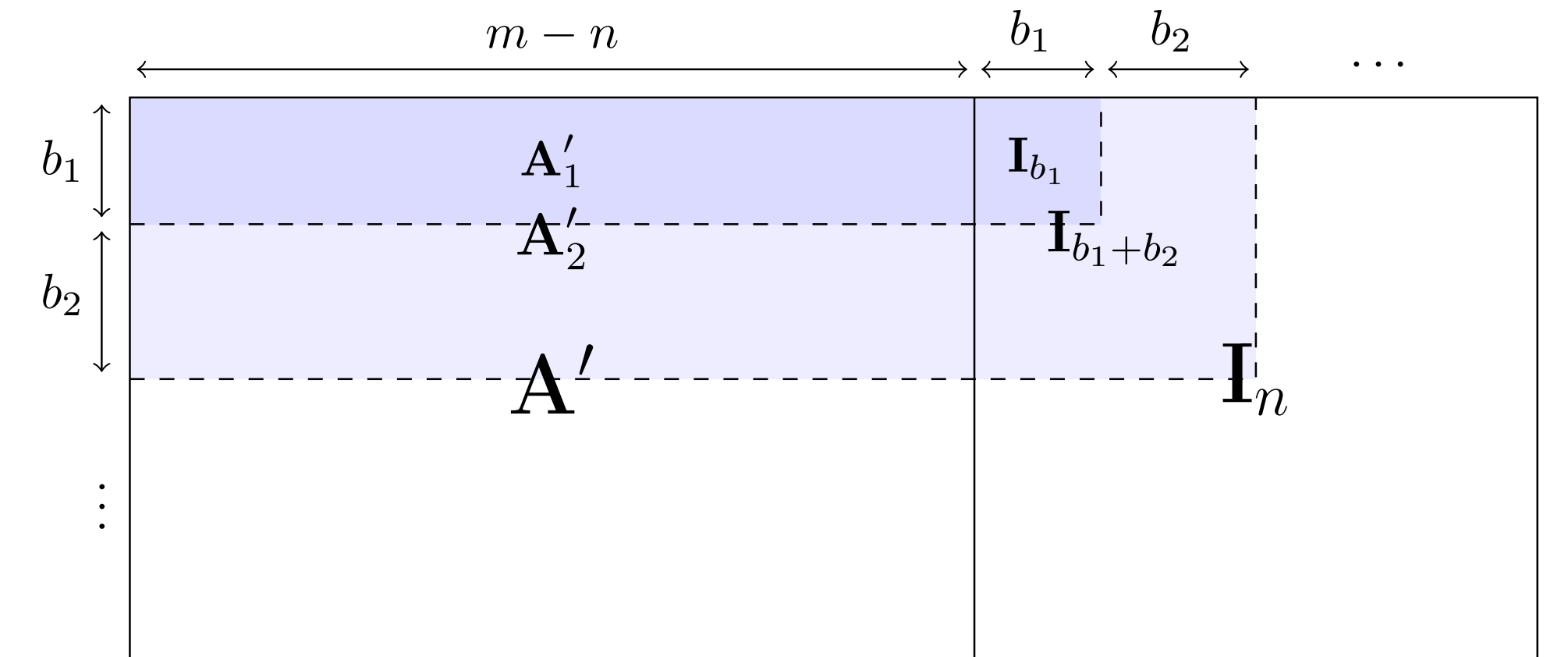


# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$
2. For  $i = 1, \dots, r$ :
  - I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$
  - II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:



# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

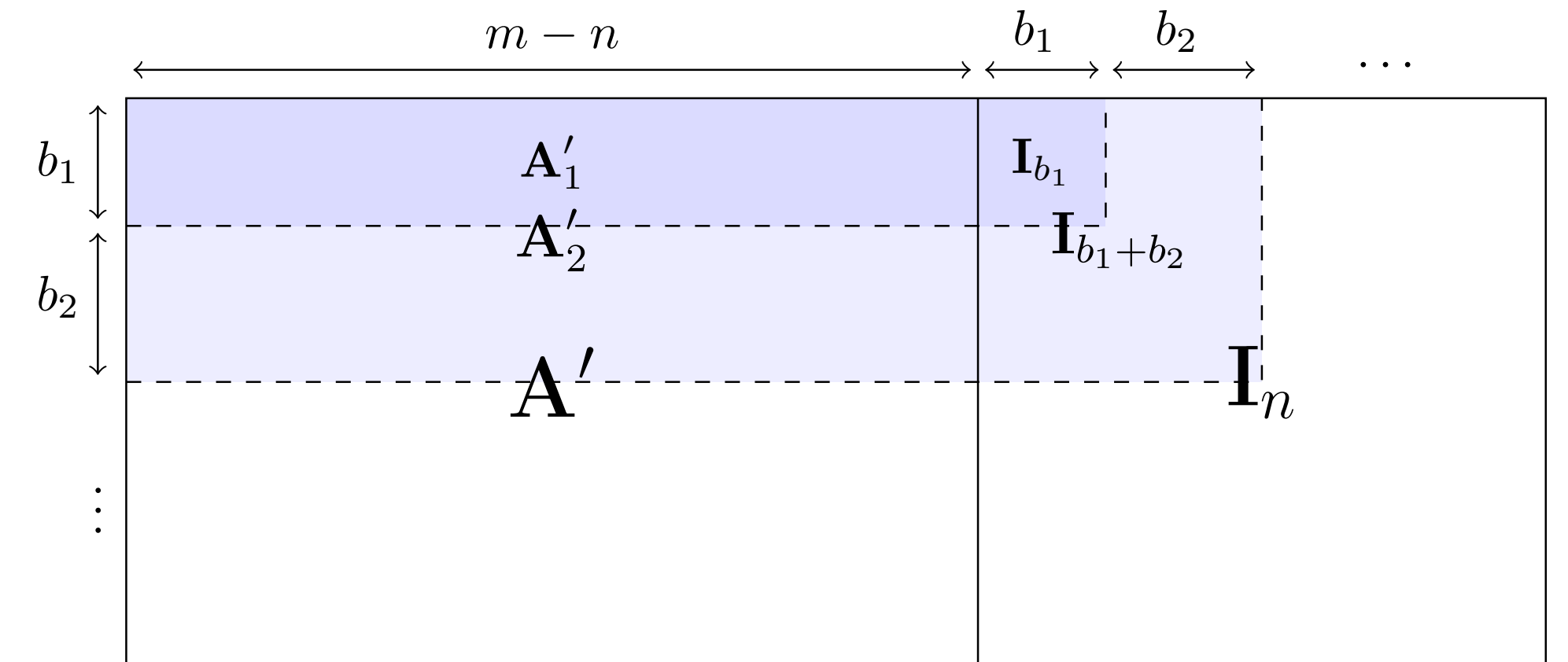
1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

▸ Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')



# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

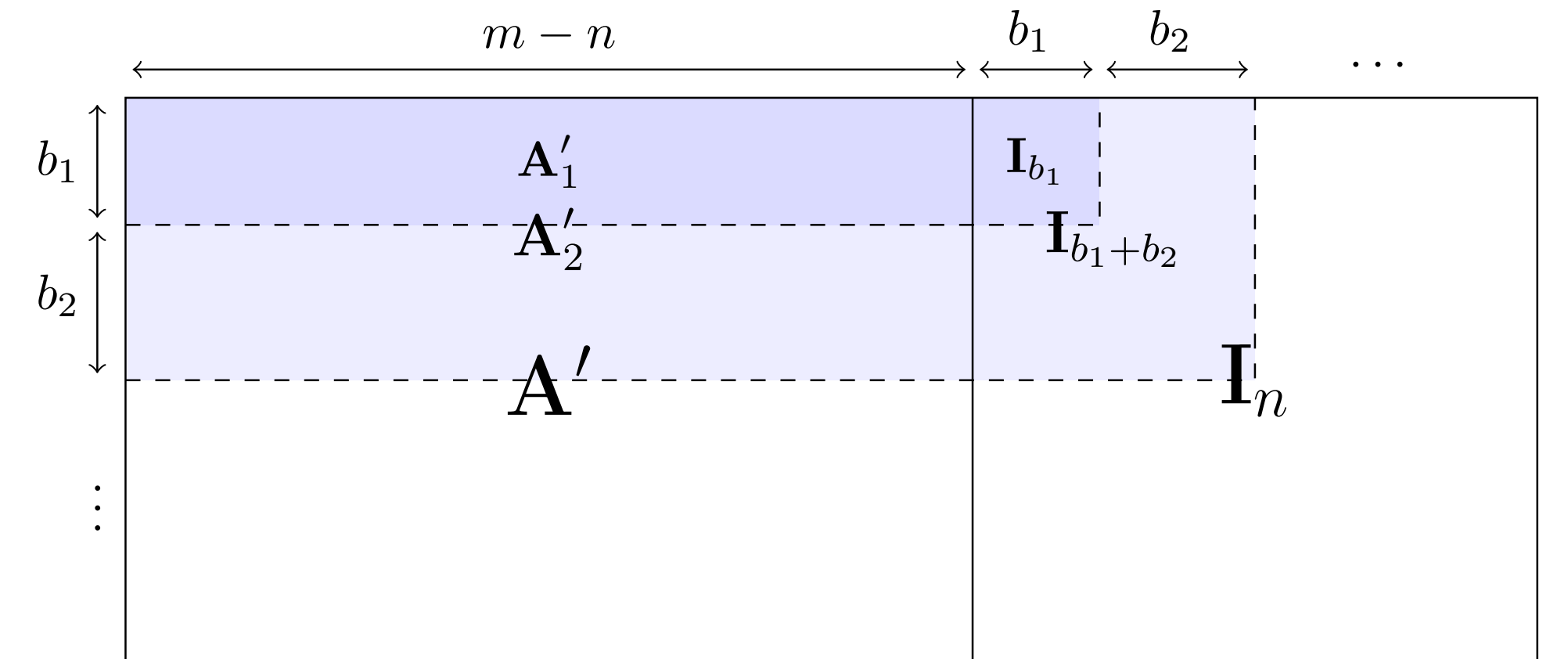
1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

- Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
- Add their difference to  $L_i$



# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

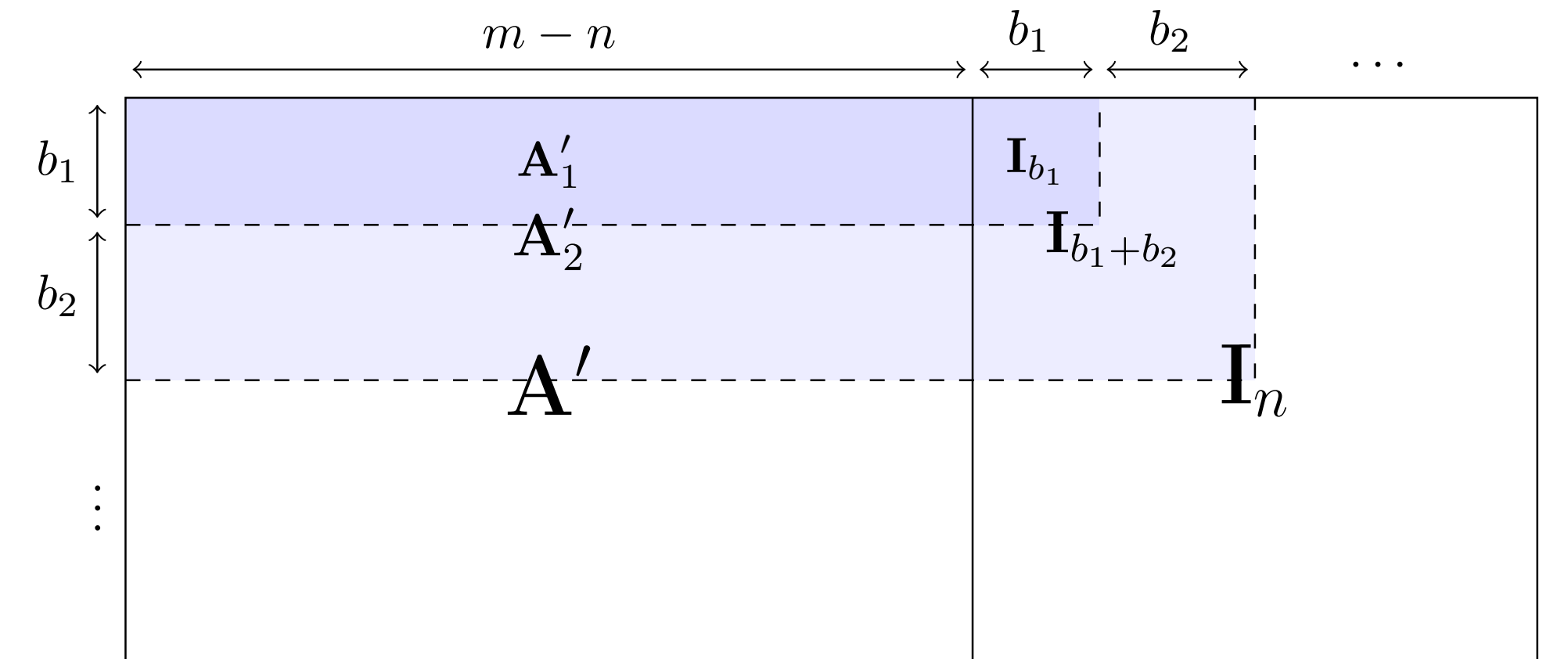
2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

- Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
- Add their difference to  $L_i$

3. Return  $L_r$



# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

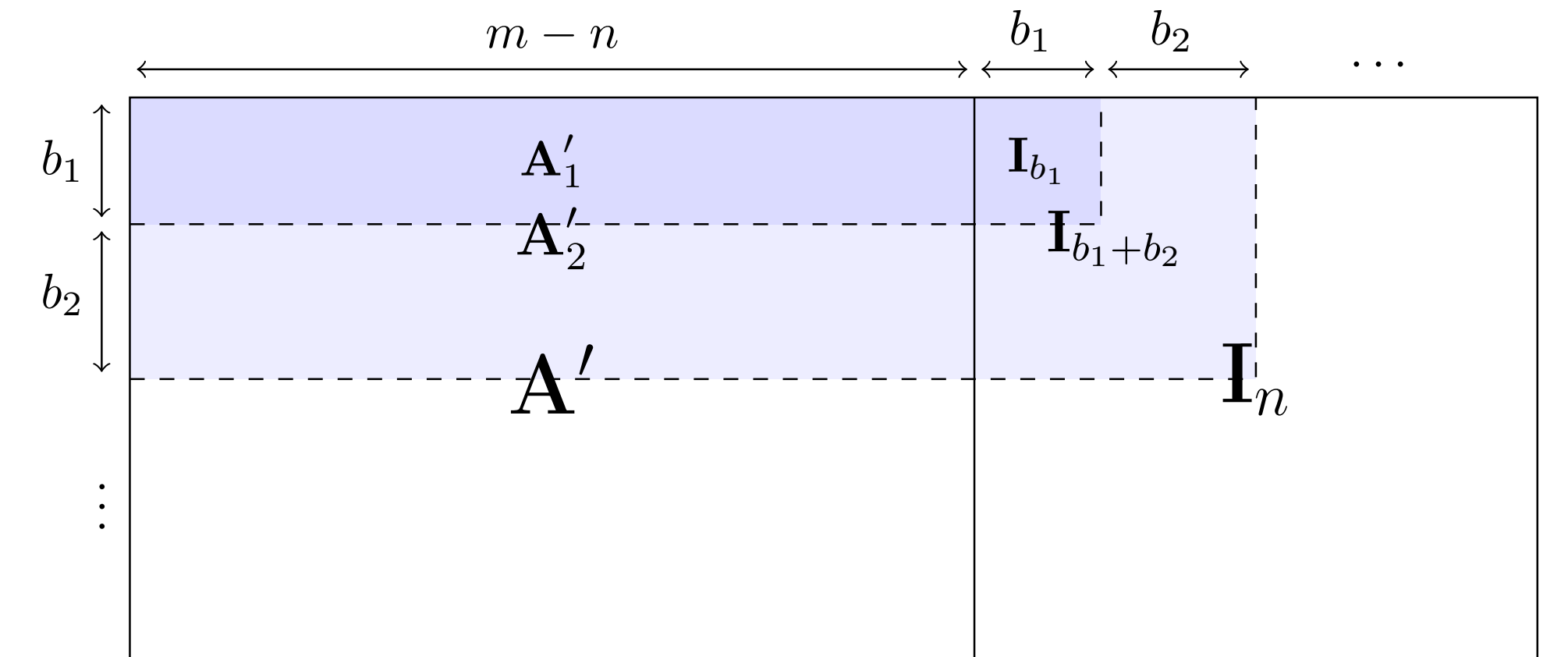
2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

- Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
- Add their difference to  $L_i$

3. Return  $L_r$



There exists a choice of parameters such that:



# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

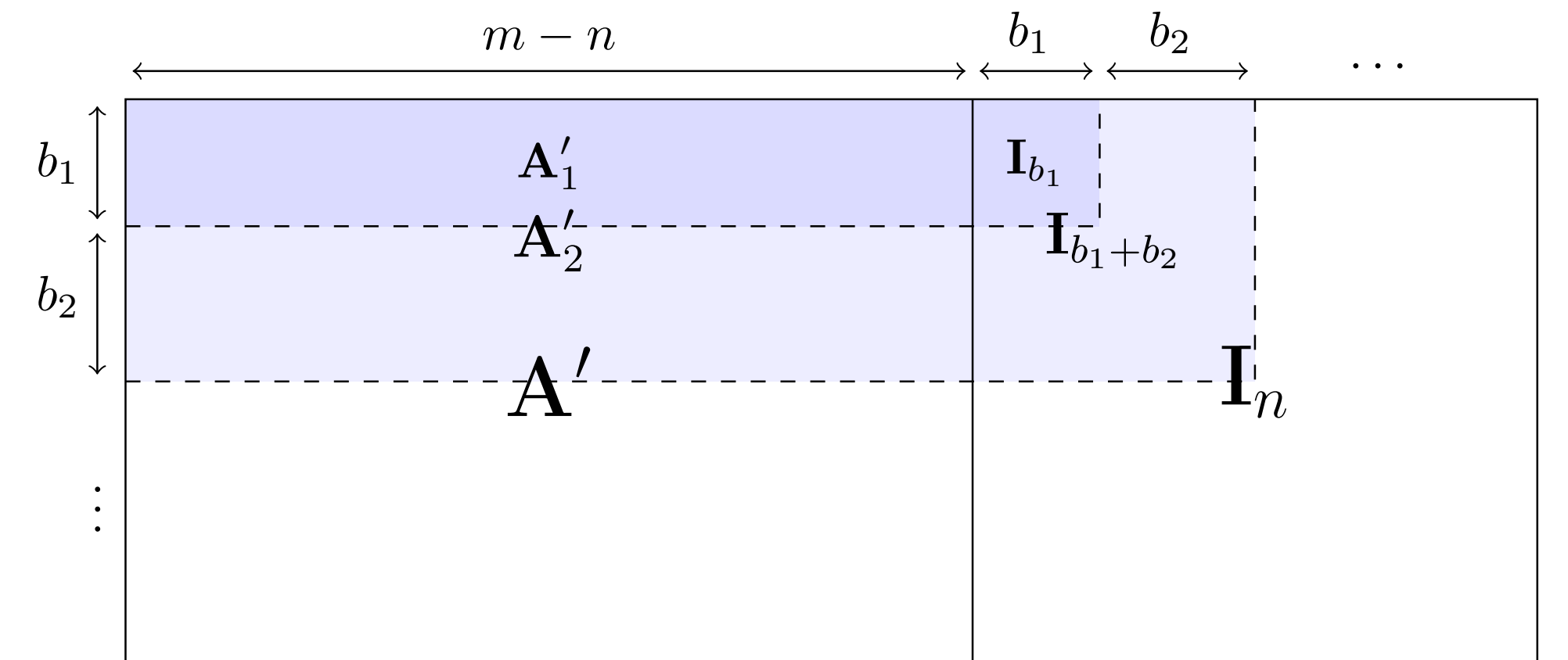
II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

- Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
- Add their difference to  $L_i$

3. Return  $L_r$

There exists a choice of parameters such that:

- ❖ Runtime is **subexponential**

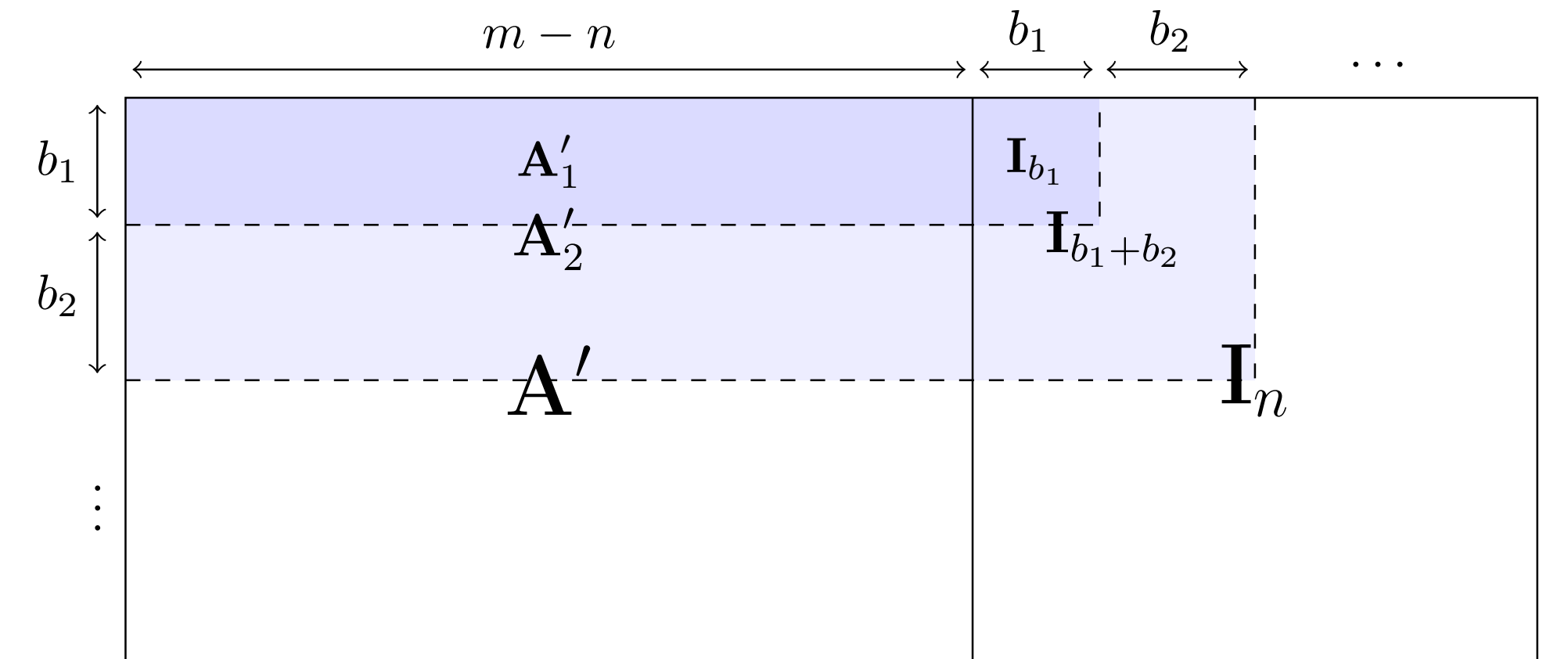


# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$
2. For  $i = 1, \dots, r$ :
  - I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$
  - II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:
    - Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
    - Add their difference to  $L_i$
3. Return  $L_r$



There exists a choice of parameters such that:

- ❖ Runtime is **subexponential**
- ❖  $L_r$  consists of solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod{q}$  that satisfy  $\|\mathbf{x}\|_\infty \leq \beta$

# Kirchner and Fouque's algorithm for SIS<sup>∞</sup>

## Algorithm:

Parameters:  $N, r, (b_i)_{i=1}^r, (p_i)_{i=1}^r$

1. Fill a list  $L_0$  with  $N$  random vectors in  $\{-1, 0, 1\}^{m-n} \subseteq \mathbb{Z}^{m-n}$

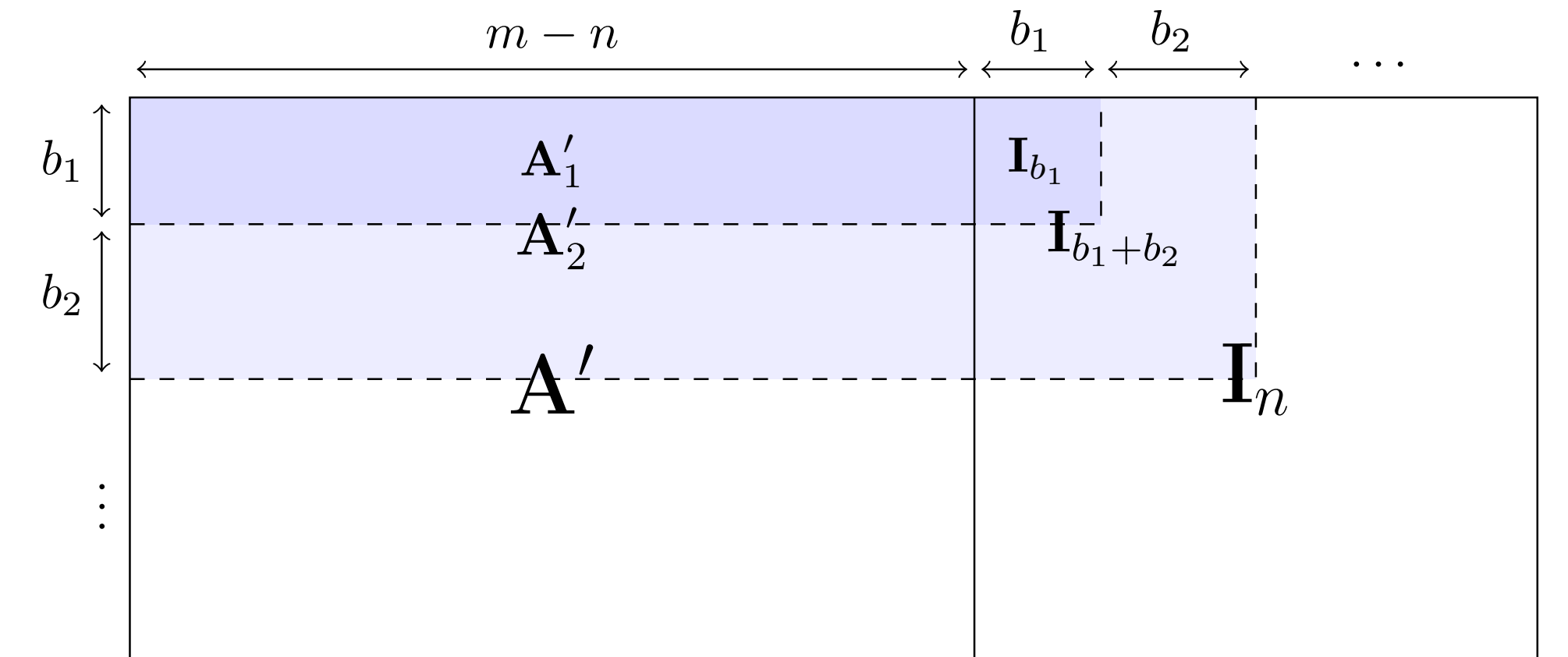
2. For  $i = 1, \dots, r$ :

I. **Lift** each  $\mathbf{x} \in L_{i-1}$  to a vector  $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$  that satisfies  $\mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0} \pmod{q}$

II. Form a new list  $L_i$  by repeatedly **combining** lifted vectors:

- Search for vectors  $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{pmatrix}$  such that  $\|\mathbf{y}_1 - \mathbf{y}_2\|_\infty \leq \frac{q}{p_i}$  (achieved via 'lazy-modulus switching')
- Add their difference to  $L_i$

3. Return  $L_r$



There exists a choice of parameters such that:

❖ Runtime is **subexponential**

❖  $L_r$  consists of solutions to  $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod{q}$  that satisfy  $\|\mathbf{x}\|_\infty \leq \beta$

*Short, but is one of them nonzero?*

# Lattice re-interpretation

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$



# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$

SIS lattice  $\Lambda_q^\perp(\mathbf{A})$



# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$

SIS lattice  $\Lambda_q^\perp(\mathbf{A})$



Re-interpretation of *lifting & combining*:

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$

SIS lattice  $\Lambda_q^\perp(\mathbf{A})$



Re-interpretation of *lifting & combining*:

- Lift** each found vector  $\mathbf{x} \in \Lambda_{i-1}$  to a vector  $\mathbf{x}'$  in the full-rank **superlattice**  $\Lambda'_i \supseteq \Lambda_i$  defined by  $\Lambda'_i = \Lambda_i + (\{0\}^{m-n+\sum_{j=1}^{i-1} b_j} \times \frac{q}{p_i} \mathbb{Z}^{b_i})$

# Lattice re-interpretation

Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$

SIS lattice  $\Lambda_q^\perp(\mathbf{A})$



Re-interpretation of *lifting & combining*:

- I. **Lift** each found vector  $\mathbf{x} \in \Lambda_{i-1}$  to a vector  $\mathbf{x}'$  in the full-rank **superlattice**  $\Lambda'_i \supseteq \Lambda_i$  defined by  $\Lambda'_i = \Lambda_i + (\{0\}^{m-n+\sum_{j=1}^{i-1} b_j} \times \frac{q}{p_i} \mathbb{Z}^{b_i})$
- II. **Combine** lifted vectors  $\mathbf{x}'_1, \mathbf{x}'_2 \in \Lambda'_i$  that belong to the same coset modulo  $\Lambda_i$ , yielding a vector  $\mathbf{x}'_1 - \mathbf{x}'_2 \in \Lambda_i$

# Lattice re-interpretation

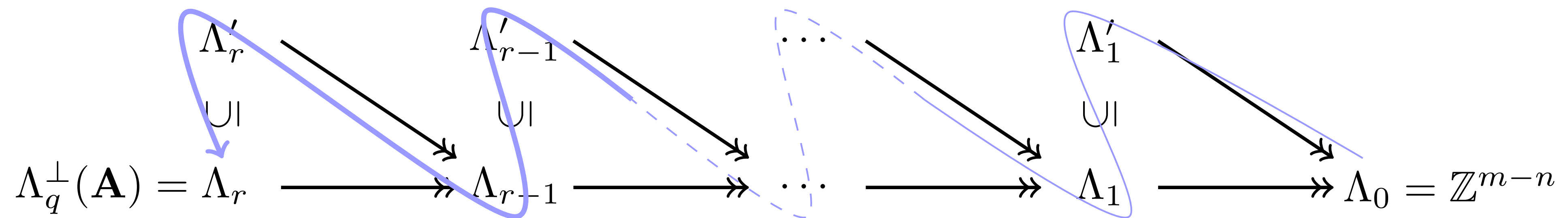
Kirchner and Fouque's SIS algorithm walks through a chain of lattices  $\Lambda_0, \Lambda_1, \dots, \Lambda_r$ :

- ❖ It starts with sampling many short random vectors in  $\Lambda_0 := \mathbb{Z}^{m-n}$
- ❖ Iteratively, by *lifting & combining* short vectors in  $\Lambda_{i-1}$ , it obtains short vectors in  $\Lambda_i := \{\mathbf{x} \in \mathbb{Z}^{m-n+\sum_{j=1}^i b_j} : \mathbf{A}_i \mathbf{x} = \mathbf{0} \bmod q\}$
- ❖ It outputs short vectors in  $\Lambda_r := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \mathbf{x} = \mathbf{0} \bmod q\}$

SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

Re-interpretation of *lifting & combining*:

- Lift** each found vector  $\mathbf{x} \in \Lambda_{i-1}$  to a vector  $\mathbf{x}'$  in the full-rank **superlattice**  $\Lambda'_i \supseteq \Lambda_i$  defined by  $\Lambda'_i = \Lambda_i + (\{0\}^{m-n+\sum_{j=1}^{i-1} b_j} \times \frac{q}{p_i} \mathbb{Z}^{b_i})$
- Combine** lifted vectors  $\mathbf{x}'_1, \mathbf{x}'_2 \in \Lambda'_i$  that belong to the same coset modulo  $\Lambda_i$ , yielding a vector  $\mathbf{x}'_1 - \mathbf{x}'_2 \in \Lambda_i$



# Our approach

# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

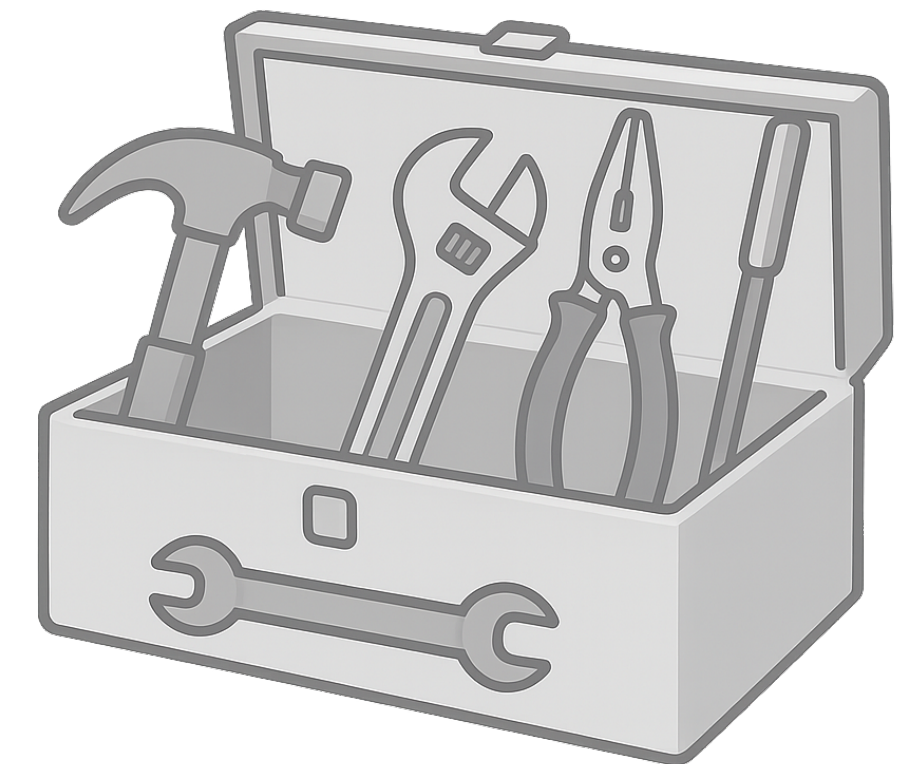
But... tracking the vector distributions seems nontrivial after the first iteration

# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

The lattice perspective opens the door to using the **discrete Gaussian toolbox**:





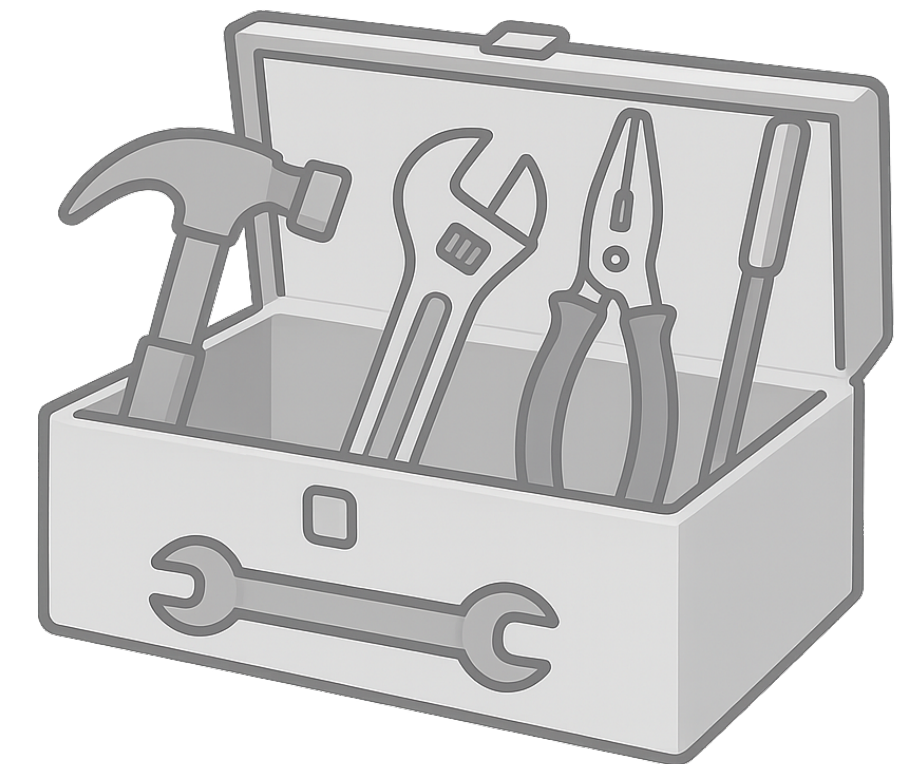
# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$



# Our approach

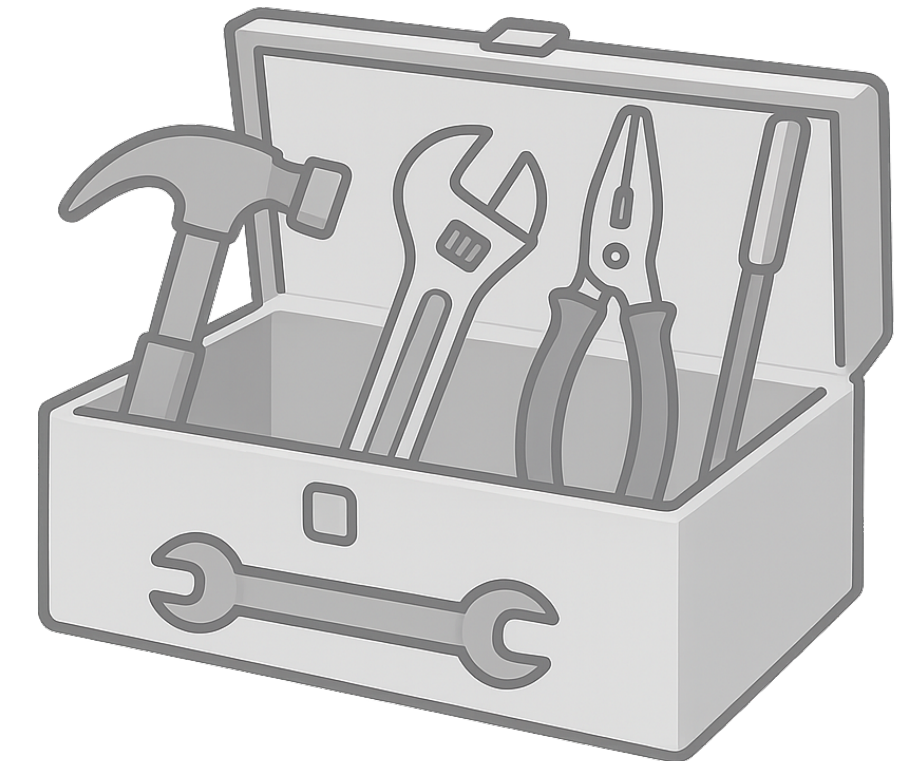
*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$
- ❖ Allows to **control the distributions** of the vectors in the lists  $L_i$

↖ *Building on a long line of previous work*



# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

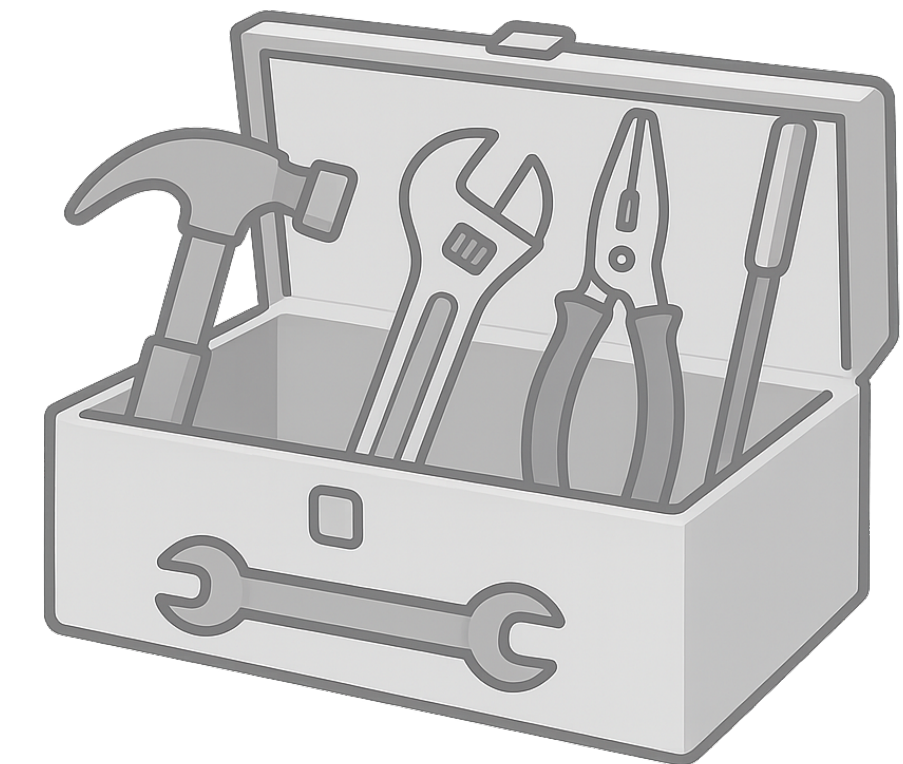
But... tracking the vector distributions seems nontrivial after the first iteration

The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$
- ❖ Allows to **control the distributions** of the vectors in the lists  $L_i$

↖ *Building on a long line of previous work*

Under certain ‘smoothing’ conditions:





# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

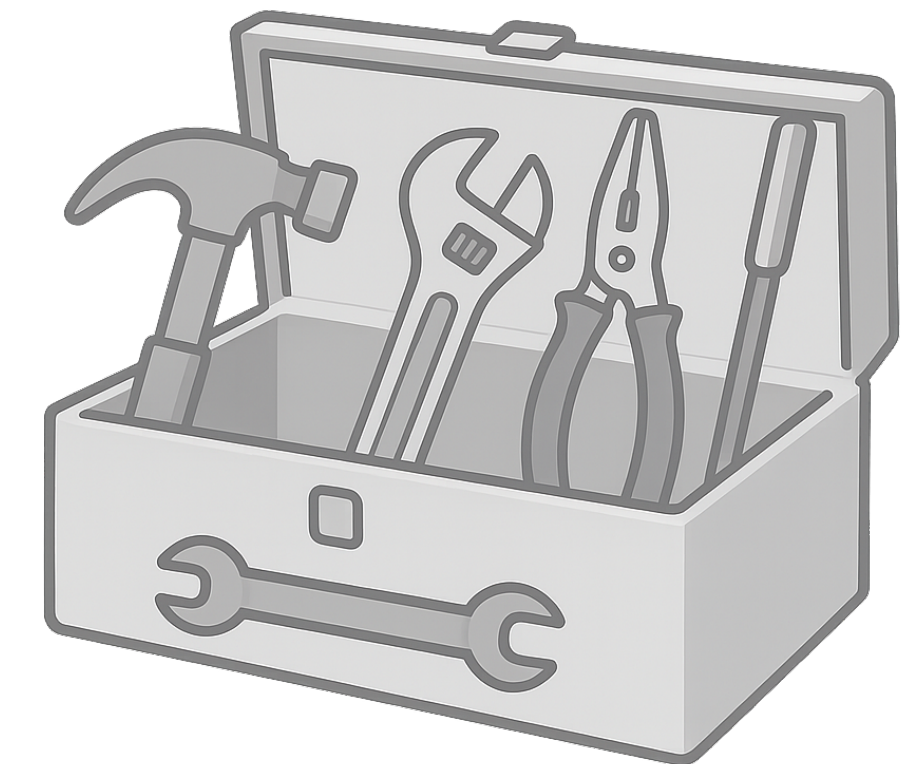
The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$
- ❖ Allows to **control the distributions** of the vectors in the lists  $L_i$

↖ *Building on a long line of previous work*

Under certain ‘smoothing’ conditions:

- ❖ The vectors in  $L_r$  are (similar to) *discrete Gaussian samples* over  $\Lambda_q^\perp(\mathbf{A})$



*The algorithm has become a Wagner-style Gaussian sampler!*

# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

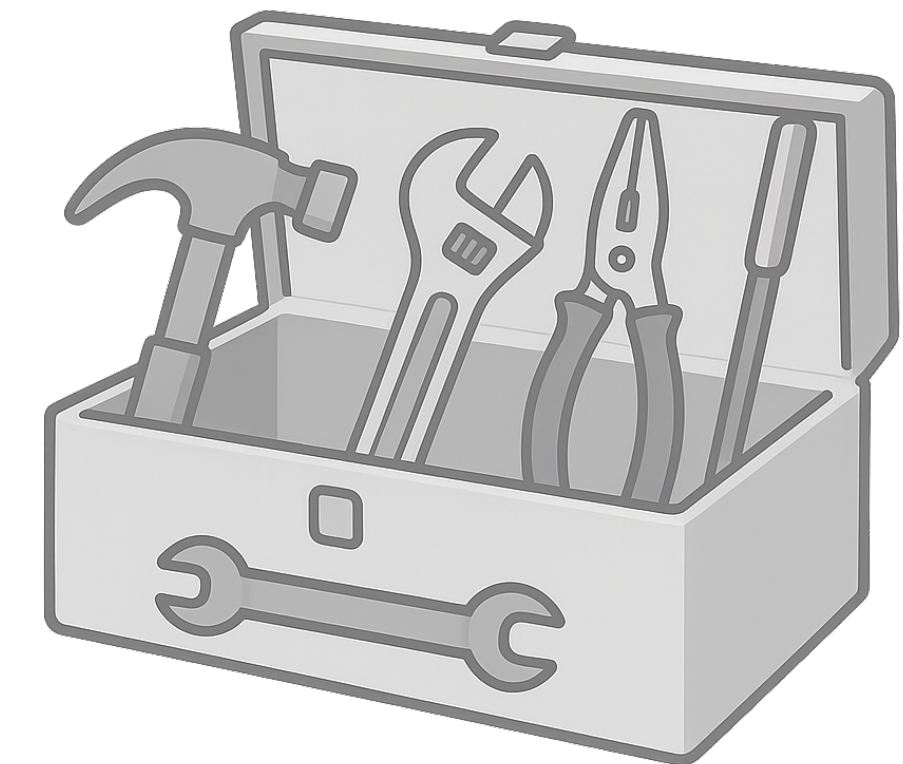
The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$
- ❖ Allows to **control the distributions** of the vectors in the lists  $L_i$

↖ *Building on a long line of previous work*

Under certain ‘smoothing’ conditions:

- ❖ The vectors in  $L_r$  are (similar to) *discrete Gaussian samples* over  $\Lambda_q^\perp(\mathbf{A})$
- ❖ With high probability, a vector in  $L_r$  is **short** (in  $\ell_2$ -norm and  $\ell_\infty$ -norm)



*The algorithm has become a Wagner-style Gaussian sampler!*

# Our approach

*Recall:* Output list  $L_r$  should contain a short and **nonzero** vector in the SIS lattice  $\Lambda_q^\perp(\mathbf{A})$

But... tracking the vector distributions seems nontrivial after the first iteration

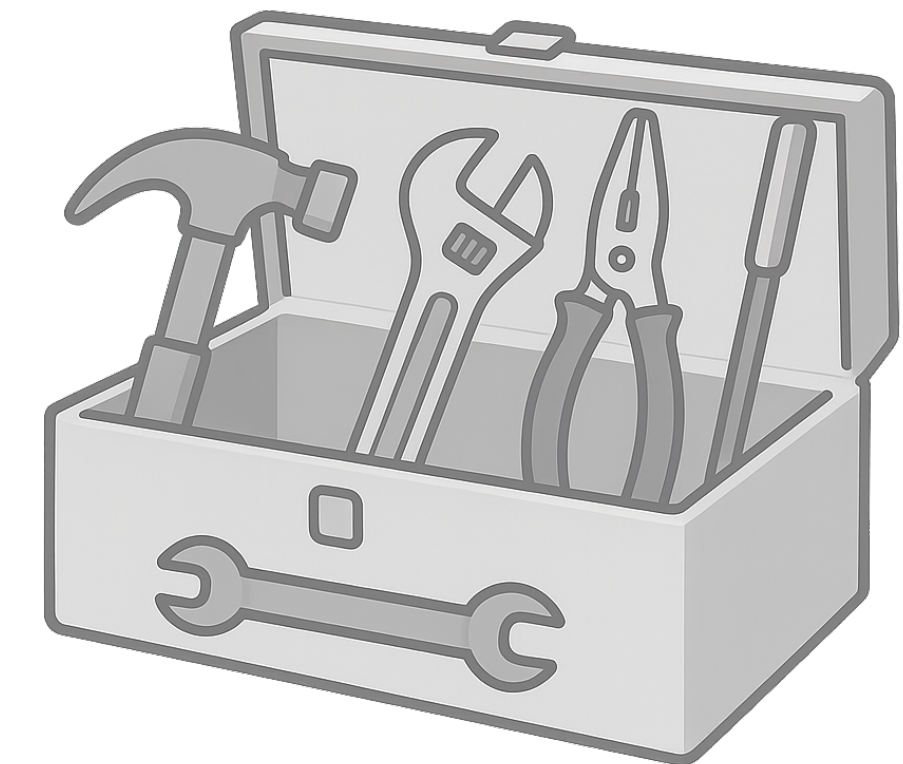
The lattice perspective opens the door to using the **discrete Gaussian toolbox**:

- ❖ Fill the initial list  $L_0$  with *discrete Gaussian samples* over  $\mathbb{Z}^{m-n}$
- ❖ Allows to **control the distributions** of the vectors in the lists  $L_i$

↖ *Building on a long line of previous work*

Under certain ‘smoothing’ conditions:

- ❖ The vectors in  $L_r$  are (similar to) *discrete Gaussian samples* over  $\Lambda_q^\perp(\mathbf{A})$
- ❖ With high probability, a vector in  $L_r$  is **short** (in  $\ell_2$ -norm and  $\ell_\infty$ -norm)
- ❖ And... **not** equal to  $\mathbf{0}$



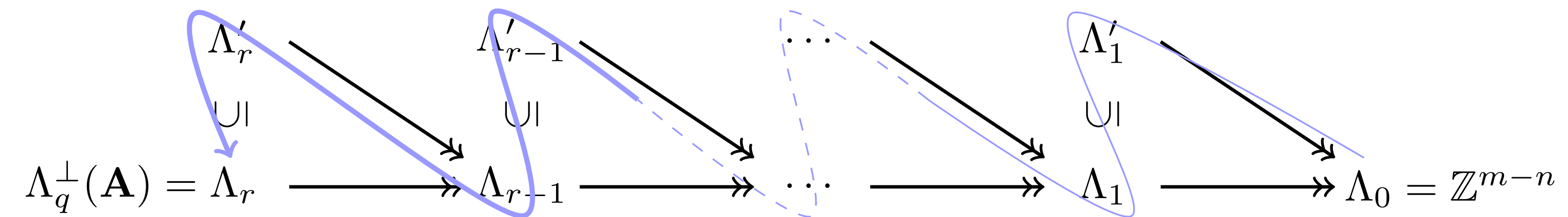
*The algorithm has become a Wagner-style Gaussian sampler!*

# Conclusion



# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:

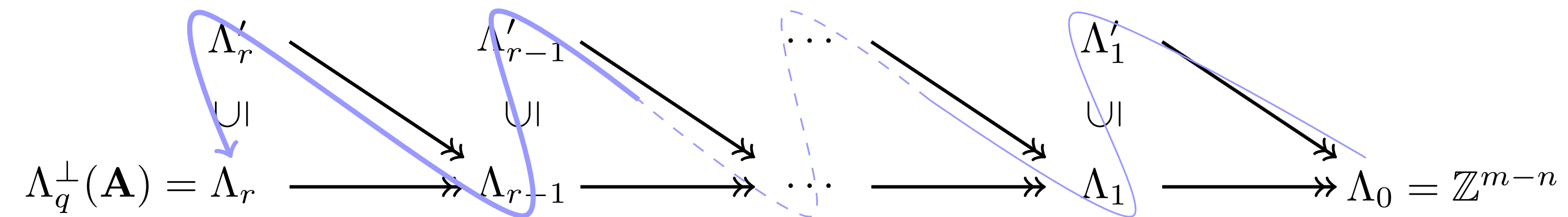


Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output



# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:

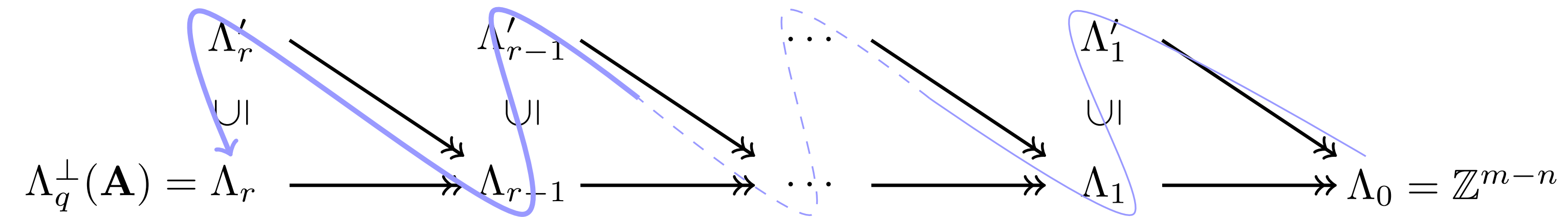


Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

Main result:

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



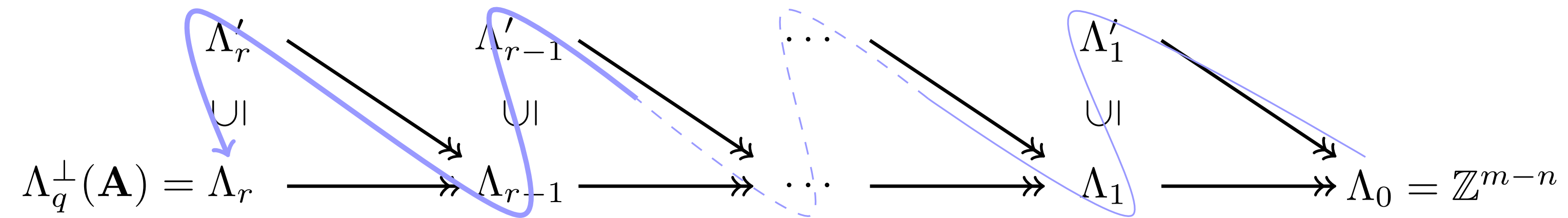
Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

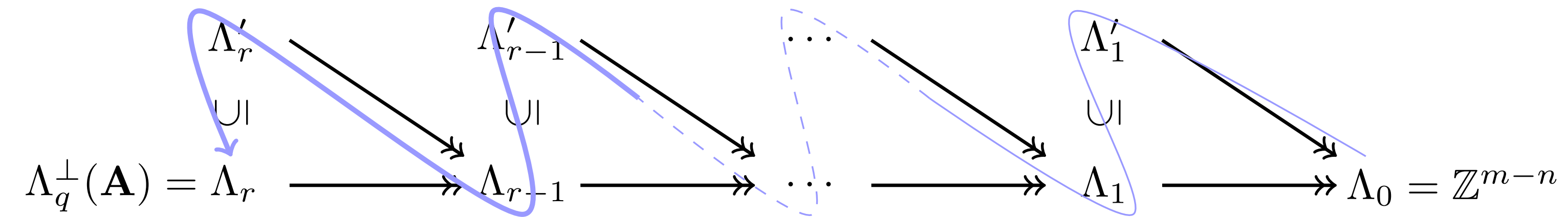
**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

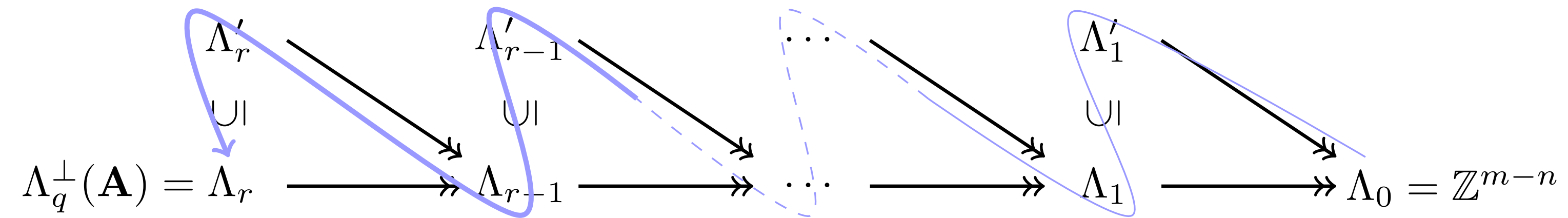
**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

- ❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )
- ❖ Asymptotic result, so ML-DSA/Dilithium is not broken

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

**Main result:**

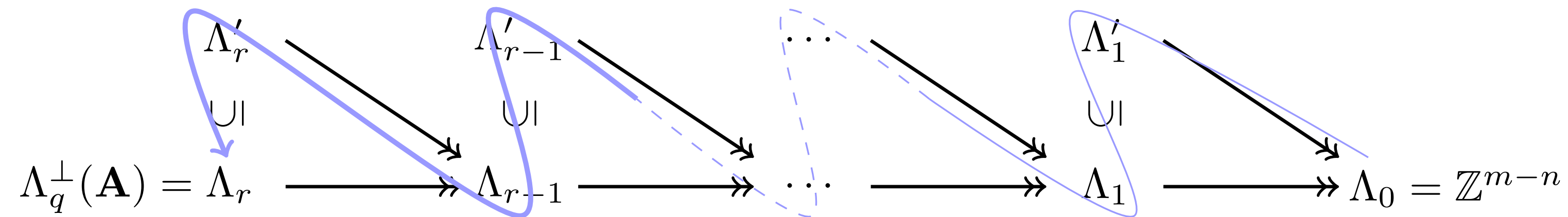
Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

- ❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )
- ❖ Asymptotic result, so ML-DSA/Dilithium is not broken

**Future work:**

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

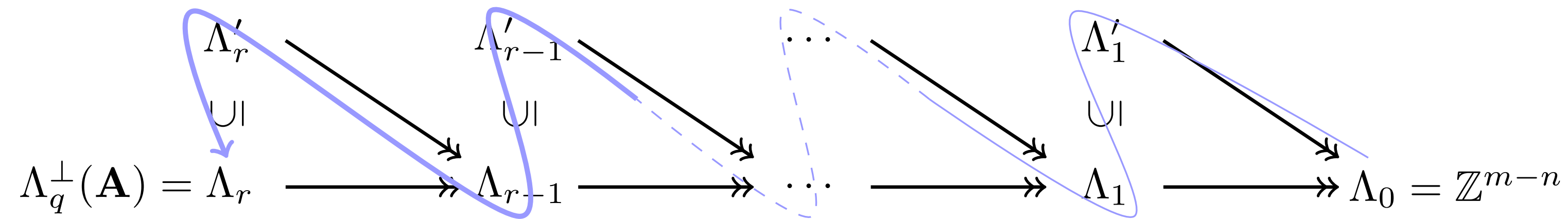
- ❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )
- ❖ Asymptotic result, so **ML-DSA/Dilithium** is not broken

**Future work:**

- ❖ Wagner-style Gaussian sampler: applications beyond SIS lattices?

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

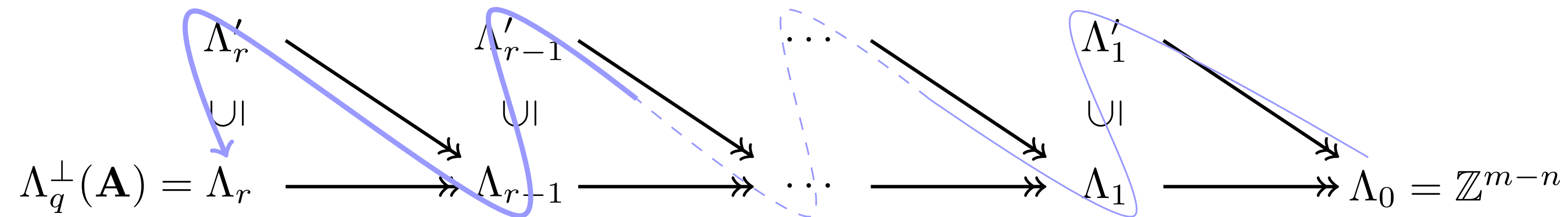
- ❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )
- ❖ Asymptotic result, so **ML-DSA/Dilithium** is not broken

**Future work:**

- ❖ Wagner-style Gaussian sampler: applications beyond SIS lattices?
- ❖ Corollary for (narrow-error) LWE?

# Conclusion

Lattice **re-interpretation** of Wagner's algorithm:



Using **discrete Gaussian tools**, we control the distributions and avoid ending up with (only)  $\mathbf{0}$  in the final output

**Main result:**

Wagner-style Gaussian sampler that solves **nontrivial** instances of  $\text{SIS}^\infty$ , Inhomogeneous-SIS, and  $\text{SIS}^\times$  in **subexponential time**

- ❖ Including parameters with  **$m = n + \omega(n/\ln \ln n)$**  (instead of  $m \approx n \ln n$ )
- ❖ Asymptotic result, so **ML-DSA/Dilithium** is not broken

**Future work:**

- ❖ Wagner-style Gaussian sampler: applications beyond SIS lattices?
- ❖ Corollary for (narrow-error) LWE?

## Thank you!

ePrint: 2025/575  
arXiv: 2503.23238