witch hunt!

demystifying lattice security

Thomas Espitau — PQShield

ICMC 2025

4. Parameter Sets

Table 1. ML-DSA parameter sets

Parameters (see Sections 6.1 and 6.2 of this document)	Values assigned by each parameter set ML-DSA-44	ML-DSA-65	ML-DSA-87
q - modulus [see §6.1]	8380417	8380417	8380417
ζ - a 512th root of unity in $\mathbb{Z}_q$ [see §7.5]	1753	1753	1753
d - # of dropped bits from $\mathbf{t}$ [see §6.1]	13	13	13
τ - # of ± 1's in polynomial c [see §6.2]	39	49	60
λ - collision strength of $\tilde{c}$ [see §6.2]	128	192	256
γ_1 - coefficient range of $\mathbf{y}$ [see §6.2]	2^{17}	2^{19}	2^{19}
γ_2 - low-order rounding range [see §6.2]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k, ℓ) - dimensions of $\mathbf{A}$ [see §6.1]	(4,4)	(6,5)	(8,7)
η - private key range [see §6.1]	2	4	2
$\beta = \tau \cdot \eta$ [see §6.2]	78	196	120
ω - max # of 1's in the hint $\mathbf{h}$ [see §6.2]	80	55	75
Challenge entropy $\log_2 \binom{256}{\tau} + \tau$ [see §6.2]	192	225	257
Repetitions (see explanation below)	4.25	5.1	3.85

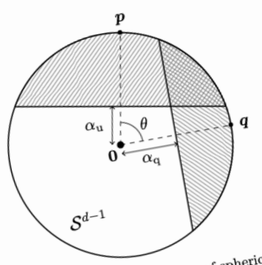
are the dimensions of the matrix $\mathbf{A}$.

These parameter sets were designed to meet certain security strength categories defined by NIST in its original Call for Proposals [26]. These security strength categories are explained further in SP 800-57, Part 1 [9].

Using this approach, security strength is not described by a single number, such as "128 bits of security." Instead, each ML-DSA parameter set is claimed to be at least as secure as a generic block cipher with a prescribed key size or a generic hash function with a prescribed output length. More precisely, it is claimed that the computational resources needed to break ML-DSA are greater than or equal to the computational resources needed to break the block cipher or hash function when these computational resources are estimated using any realistic model of computation. Different models of computation can be more or less realistic and, accordingly, lead to more or less accurate estimates of security strength. Some commonly studied models are discussed in [27].

Concretely, the parameter set ML-DSA-44 is claimed to be in security strength category 2, ML-DSA-65 is claimed to be in category 3, and ML-DSA-87 is claimed to be in category 5 [6]. For additional discussion of the security strength of MLWE-based cryptosystems, see [28].

The sizes of keys and signatures that correspond to each parameter set are given in Table 2. Certain optimizations are possible when storing ML-DSA public and private keys. If additional space is available, one can precompute and store $\hat{\mathbf{A}}$ to speed up signing and verifying. Alternatively, if one wants to reduce



$\Pr_{u \sim S^{d-1}}[u \in \text{Update}(p)] \propto \mathcal{C}(\alpha_u)$

$\Pr_{u \sim S^{d-1}}[u \in \text{Update}(p) \cap \text{Query}(q)] \propto \mathcal{W}(\alpha_q, \alpha_u, \theta)$

$\Pr_{u \sim S^{d-1}}[u \in \text{Query}(q)] \propto \mathcal{C}(\alpha_q)$

Figure 1 The geometry of spherical filters. A vector p is inserted into a filter u with probability proportional to $\mathcal{C}(\alpha_u)$, over the randomness of sampling u at random from S^{d-1}. A filter u is queried for near neighbors for q with probability $\mathcal{C}(\alpha_q)$. A vector p at angle θ from q is found as a candidate nearest neighbor in one of the filters with probability proportional to $\mathcal{W}(\alpha_q, \alpha_u, \theta)$.

To balance the query costs of (1) finding the $f \cdot \mathcal{C}(\alpha_q)$ relevant filters, and (2) going through $f \cdot n \cdot \mathcal{C}(\alpha_q) \cdot \mathcal{C}(\alpha_u)$ false positives, we choose α_u such that $n \cdot \mathcal{C}(\alpha_u) = 1$, i.e. we set $\alpha_u = \sqrt{1 - n^{-2/d}}$. We further set $\alpha_q = \beta \alpha_u$, so that only β remains to be chosen, and query/update tradeoff: $\beta < 1$ leads to a better space complexity, and $\beta > 1$ complexity. For $\beta = 1$ we get the results from [9].

complexity scales as $n \cdot f \cdot \mathcal{C}(\alpha_u) \sim n$, which using β would only lead

▶ **Theorem 4** (Near neighbor using spherical locality-sensitive filters with [illegible] the θ-near neighbor problem with update and query exponents ρ_u, ρ_q (5)

$$\rho_u = \log\left[1 - \left(1 - n^{-2/d}\right)\frac{1 + \beta^2 - 2\beta\cos\theta}{\sin^2\theta}\right] / \log(n^{-2/d}) - 1, \quad (6)$$

$$\rho_q = \log\left[1 - \left(1 - n^{-2/d}\right)\frac{1 + \beta^2 - 2\beta\cos\theta}{\sin^2\theta}\right] / \log(n^{-2/d}) \quad (7)$$

$$- \log\left[1 - \left(1 - n^{-2/d}\right)\beta^2\right] / \log(n^{-2/d}).$$

This data structure requires $n^{1+\rho_u+o(1)}$ memory, can be initialized in time $n^{1+\rho_u+o(1)}$, allows for updates in time $n^{\rho_u+o(1)}$, and answers queries in time $n^{\rho_q+o(1)}$.

In the limit of $n^{1/d} = 1 + \varepsilon$ with $\varepsilon \to 0$, corresponding to random data sets in the sparse regime, a Taylor expansion around $\varepsilon = 0$ of ρ_q and ρ_u yields:

$$\rho_u = \frac{(\beta - \cos\theta)^2}{\sin^2\theta} + o(1), \qquad \rho_q = \frac{(1 - \beta\cos\theta)^2}{\sin^2\theta} + o(1). \quad (8)$$

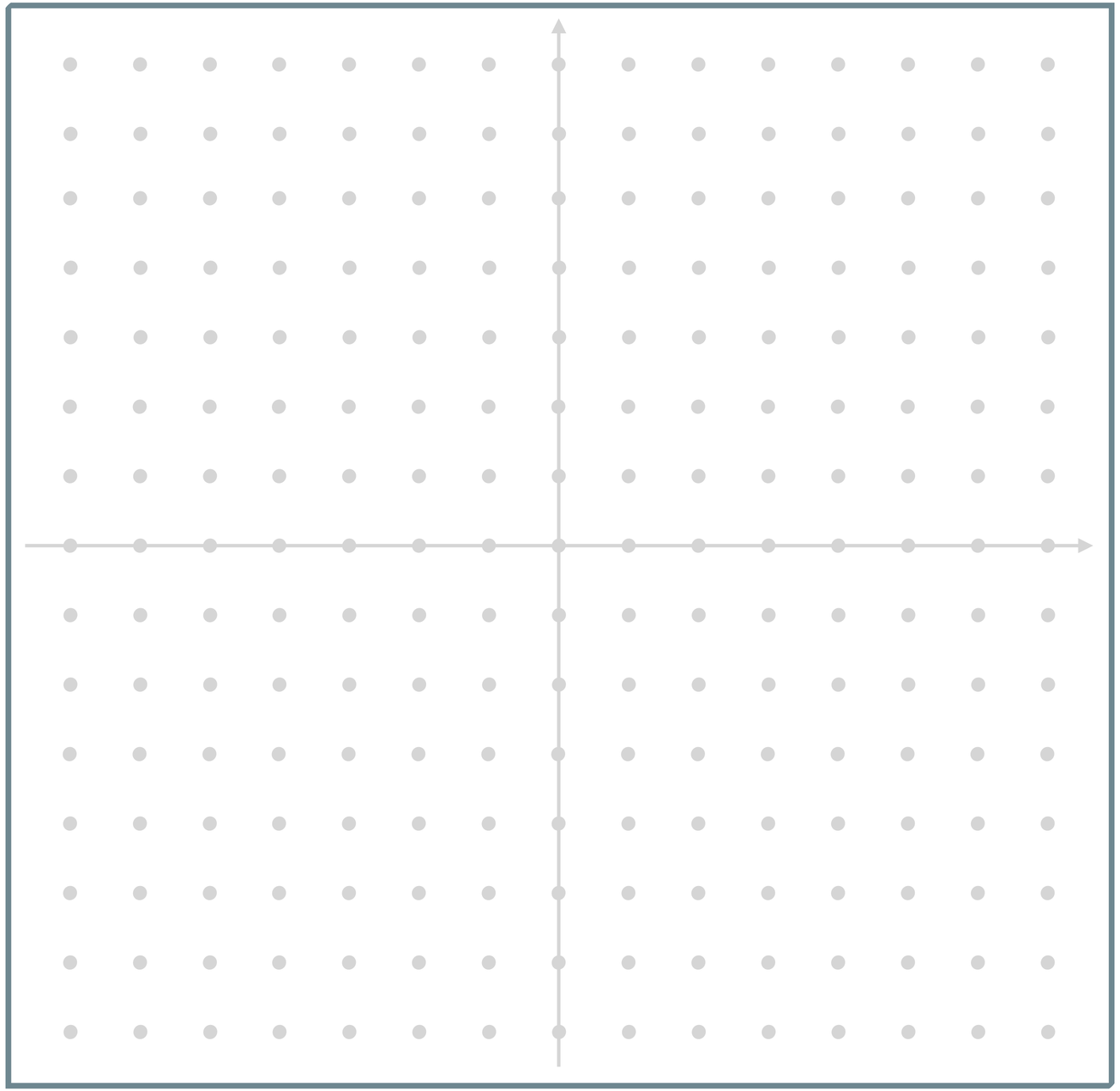
This leads to the concise defining equation $\sqrt{\rho_q} + \cos\theta\sqrt{\rho_u} = \sin\theta$ for the sparse regime, which is equivalent to [5, Equation (1)] after substituting $\cos\theta = 1 - 1/c^2$. This also clearly shows how to set β to minimize ρ_u or ρ_q in the sparse regime.

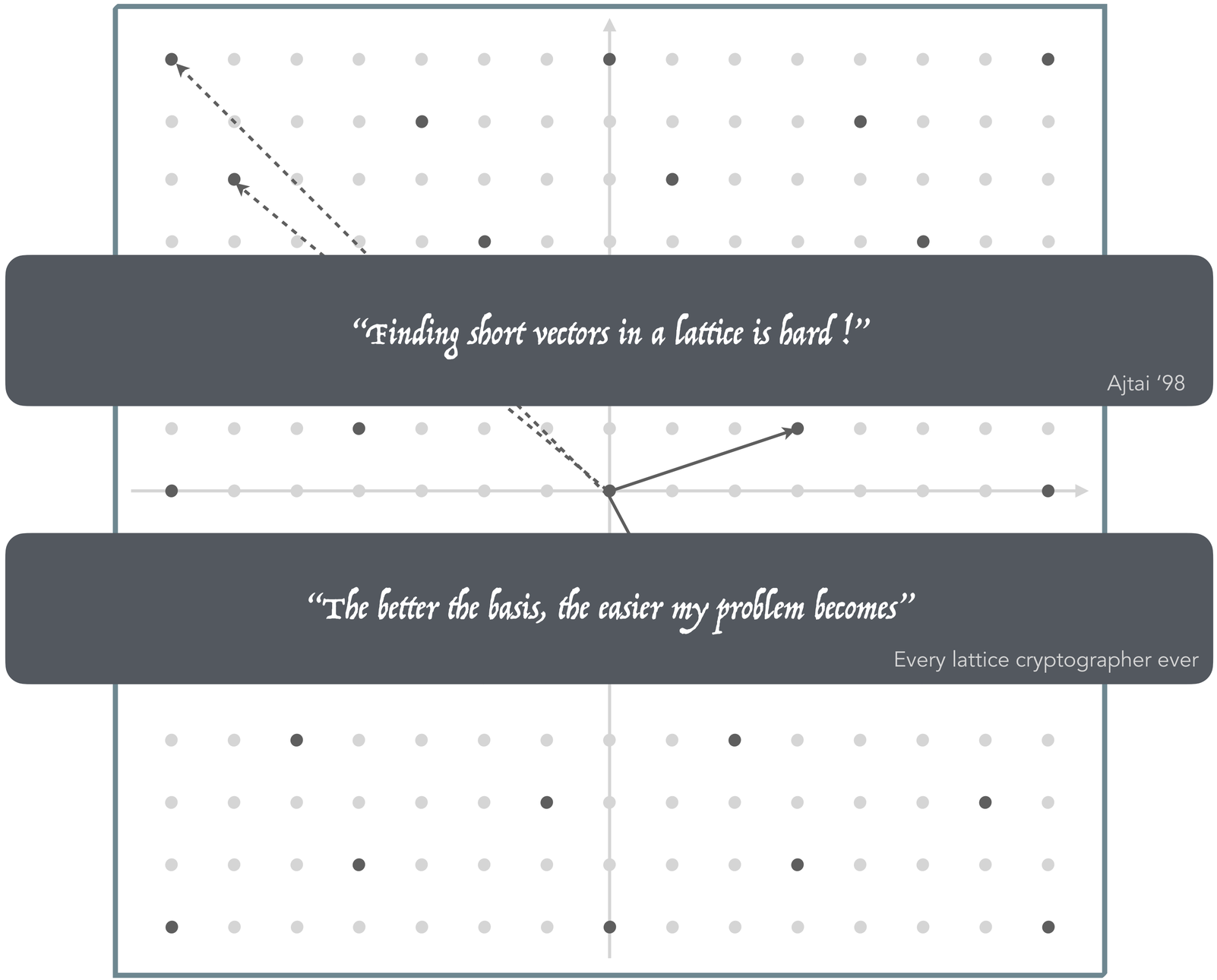
CRYSTALS-Dilithium

What kind of black magic is that ? !

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf

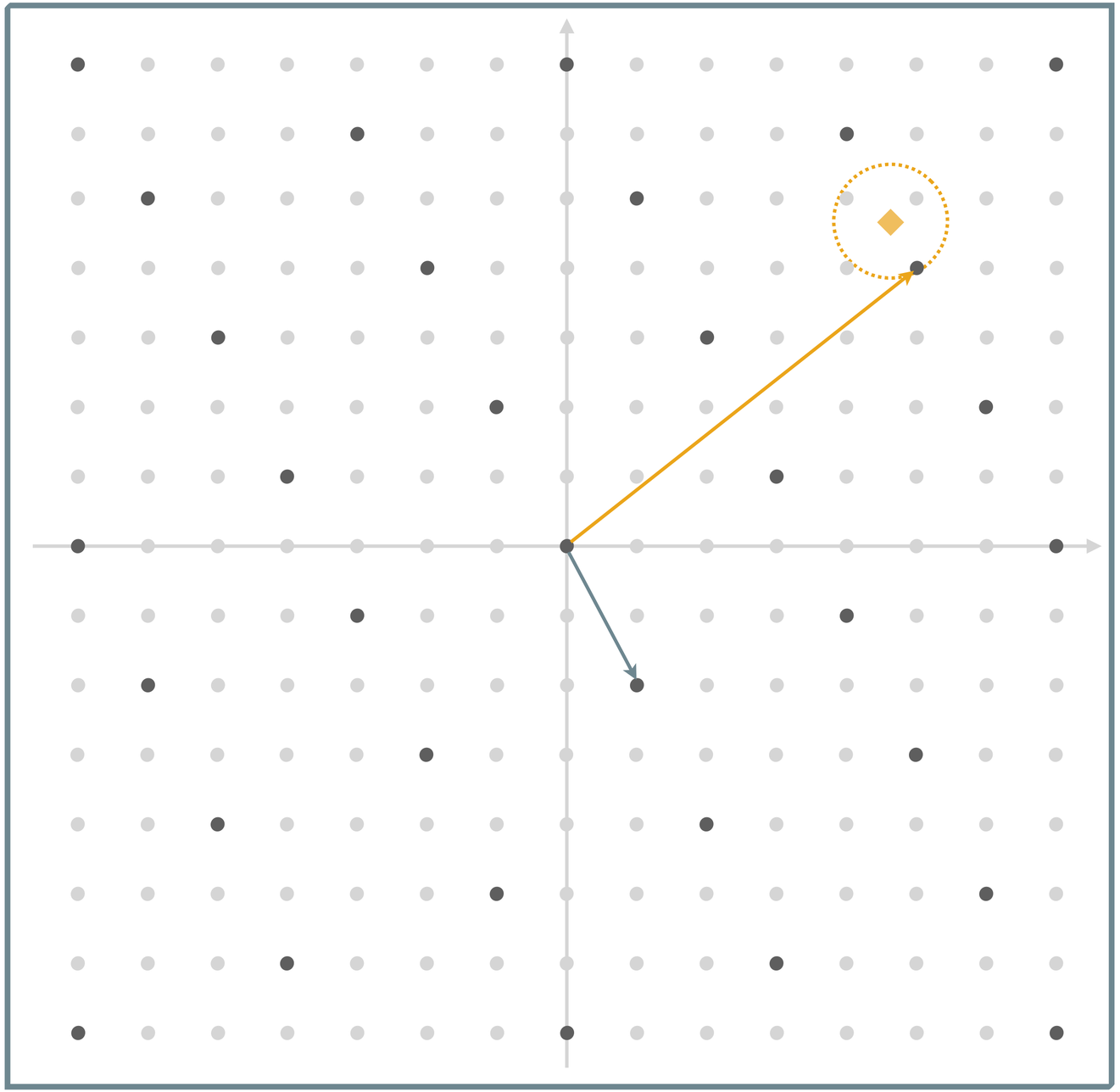
What is a lattice ?



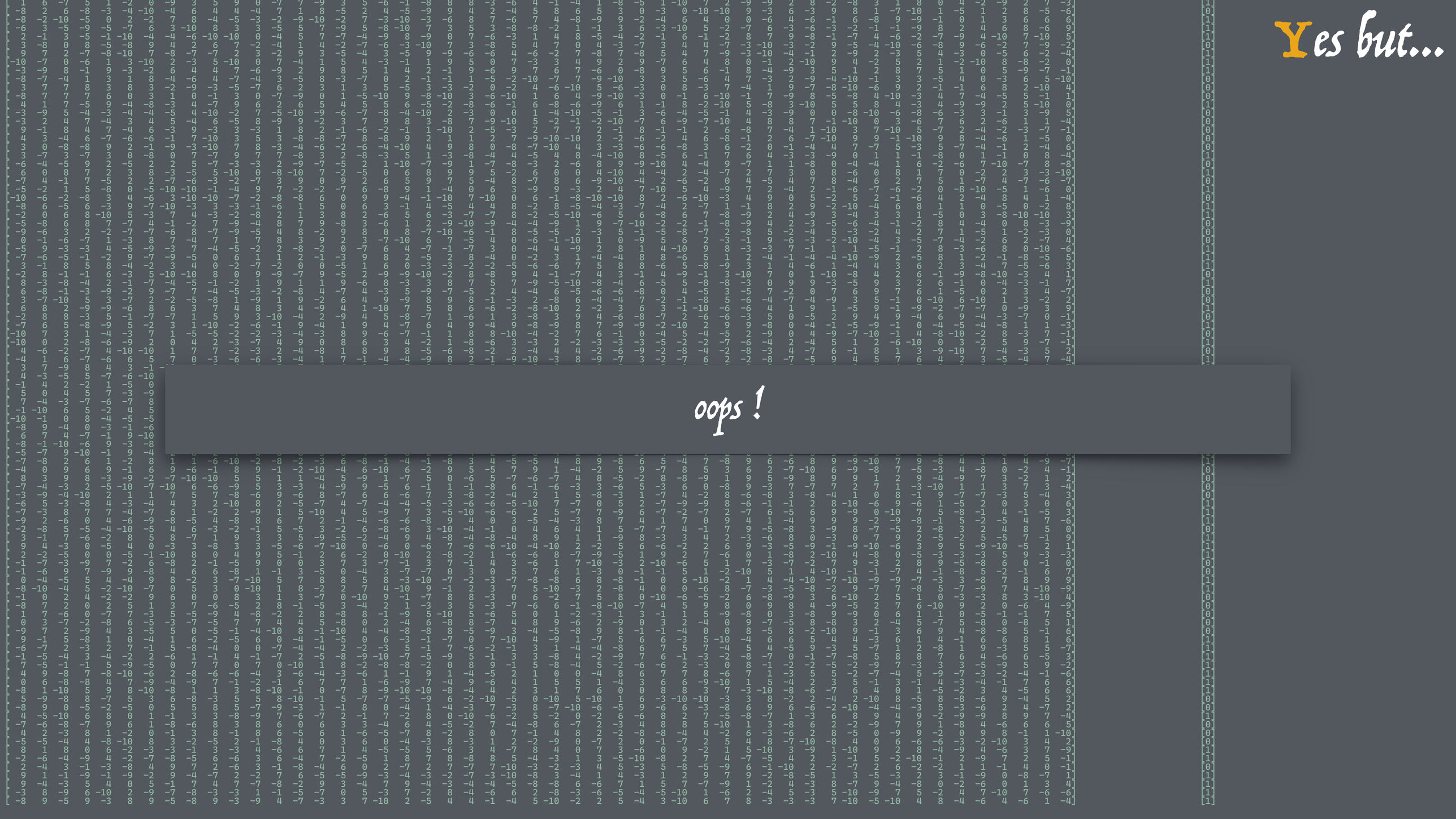

"Finding short vectors in a lattice is hard !"
Ajtai '98
"The better the basis, the easier my problem becomes"
Every lattice cryptographer ever

What are the main problems ?

Shortest vector problem
(SVP)



Closest vector problem
(CVP)


Yes but...
oops !

A more cryptographic problem

$$(1 \quad 5)\begin{pmatrix} u \\ v \end{pmatrix} = 0 \pmod 7$$

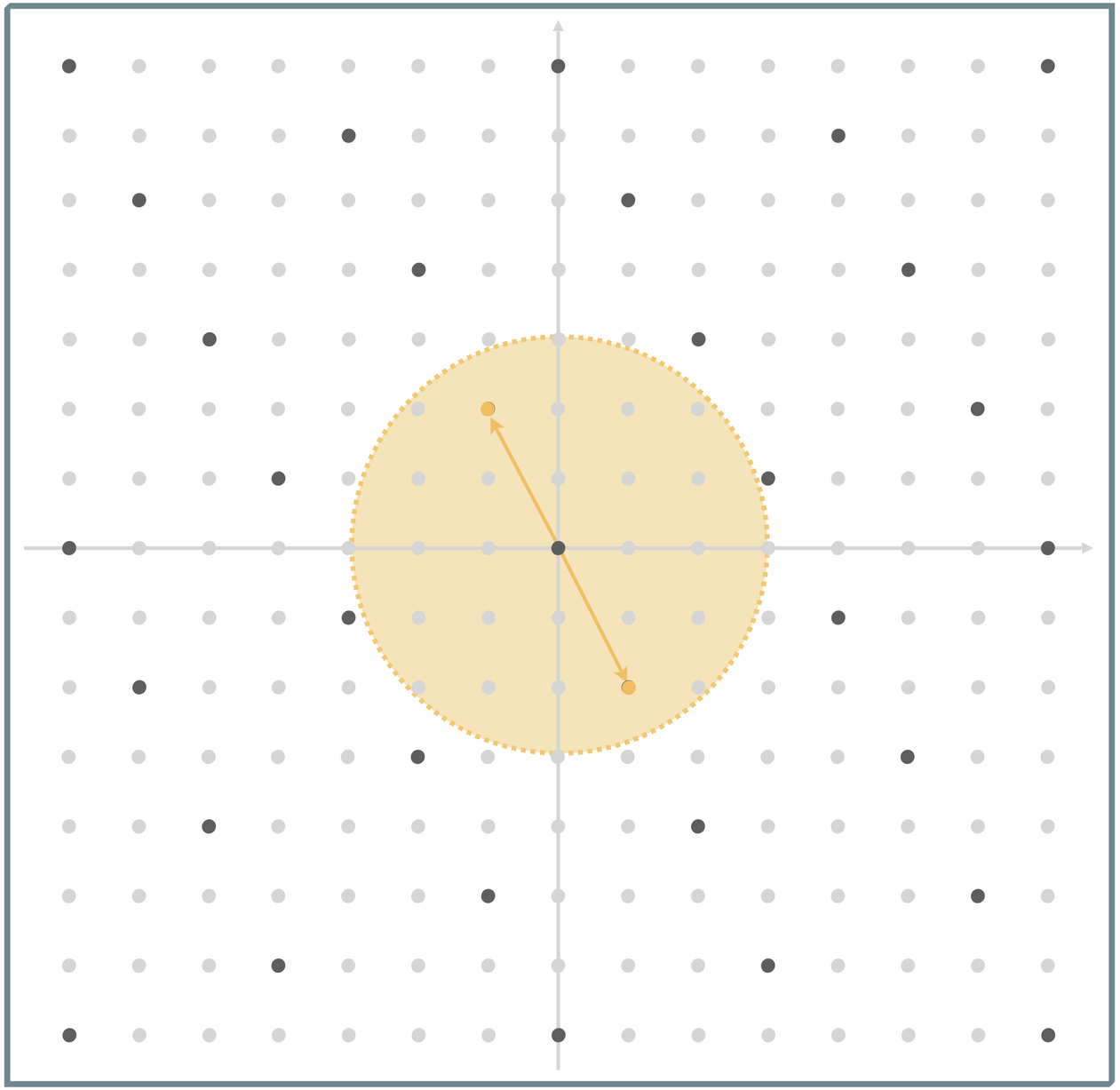
$(5 \quad -1)$ $(-5 \quad 1)$

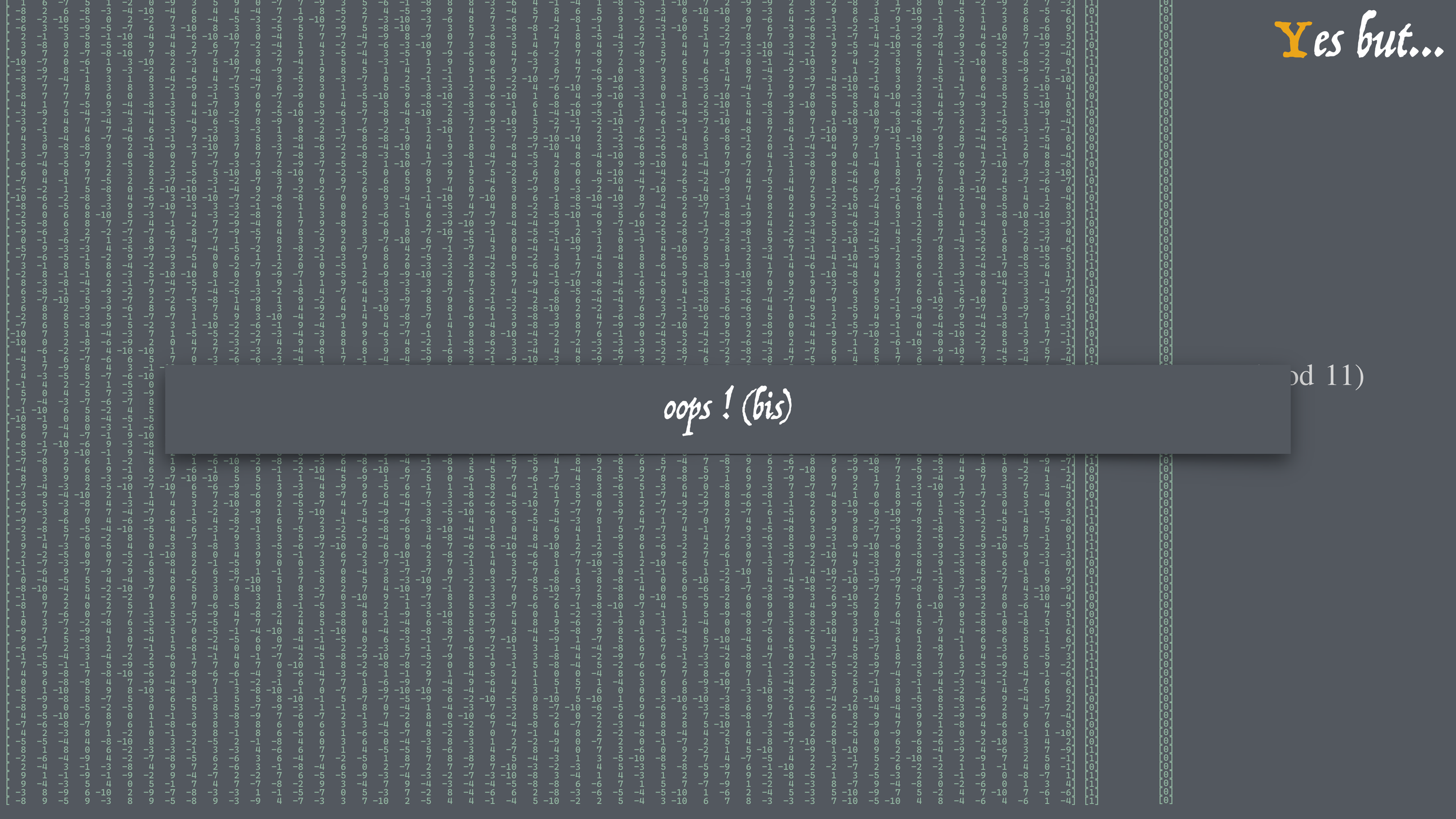
small $(2 \quad 1)$ $(-2 \quad -1)$

$(10 \quad -2)$ $(-10 \quad 2)$

The set of solutions is a lattice !

$(24 \quad -2)$ $(-16 \quad 6)$




Yes but...
oops ! (bis)

SIS

$$A \cdot \underset{\|x\| \le \eta}{x} = 0 \pmod q$$

NTRU

$$\left\{ \begin{pmatrix} u \\ v \end{pmatrix} \;\middle|\; \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} h \\ -1 \end{pmatrix} = 0 \pmod q \right\}$$

LWE

$$b = A \cdot s + e \pmod q$$

SIS

svp in

$$\left\{ x \;\middle|\; A \cdot x = 0 \pmod{q} \right\}$$

NTRU

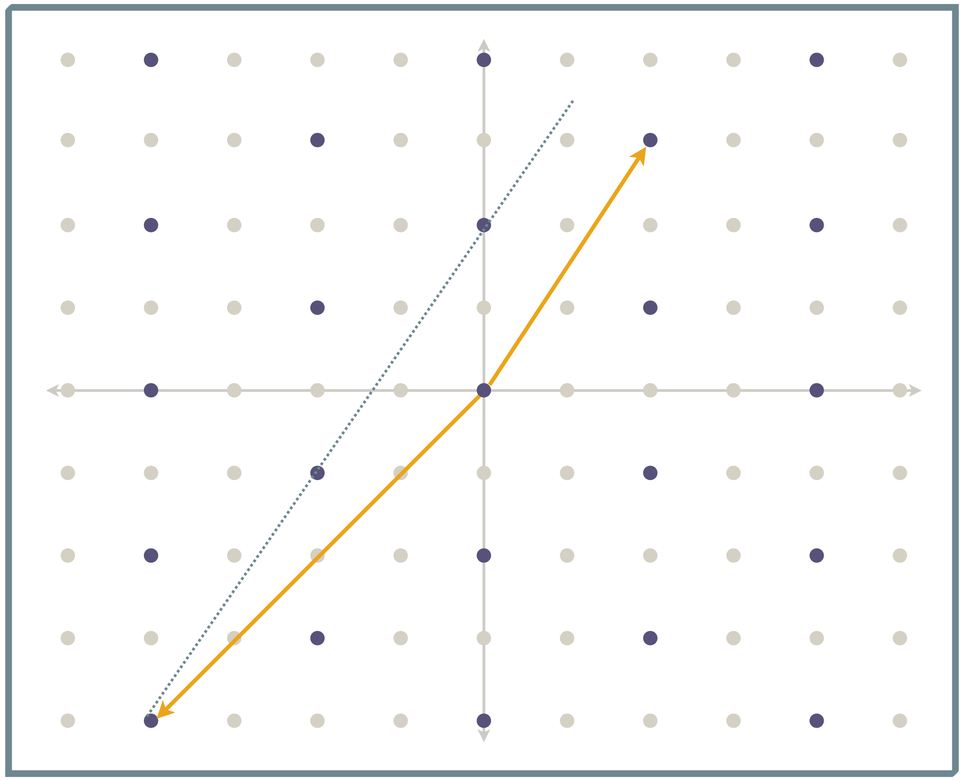
svp in $\left\{ \begin{pmatrix} u \\ v \end{pmatrix} \;\middle|\; \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} h \\ -1 \end{pmatrix} = 0 \pmod{q} \right\}$

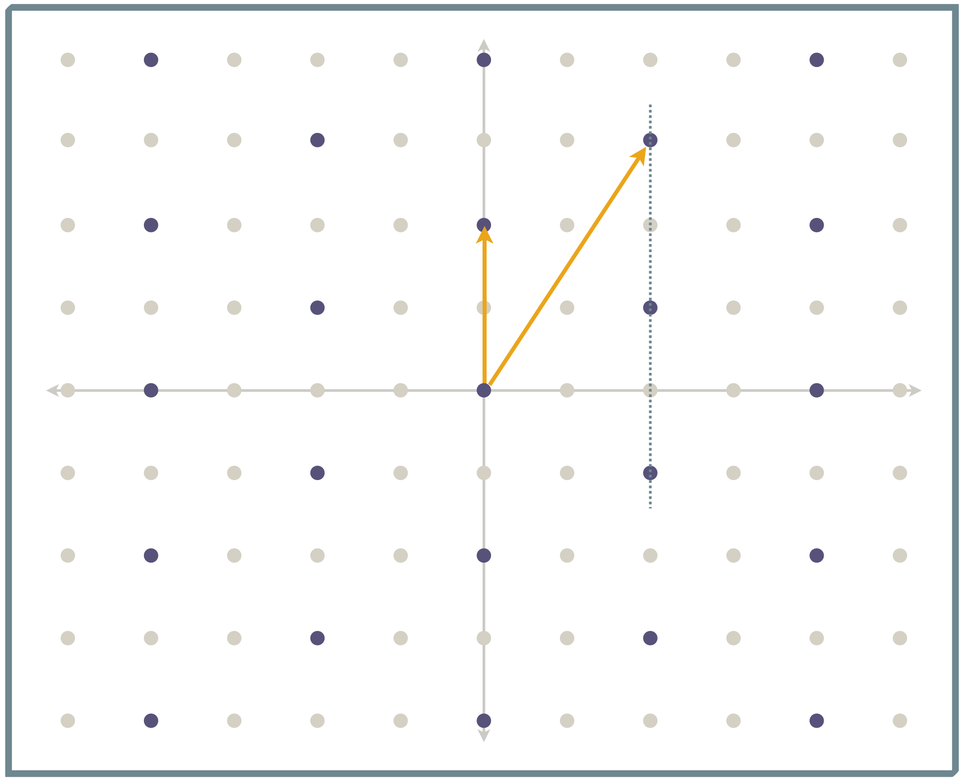
cvp in

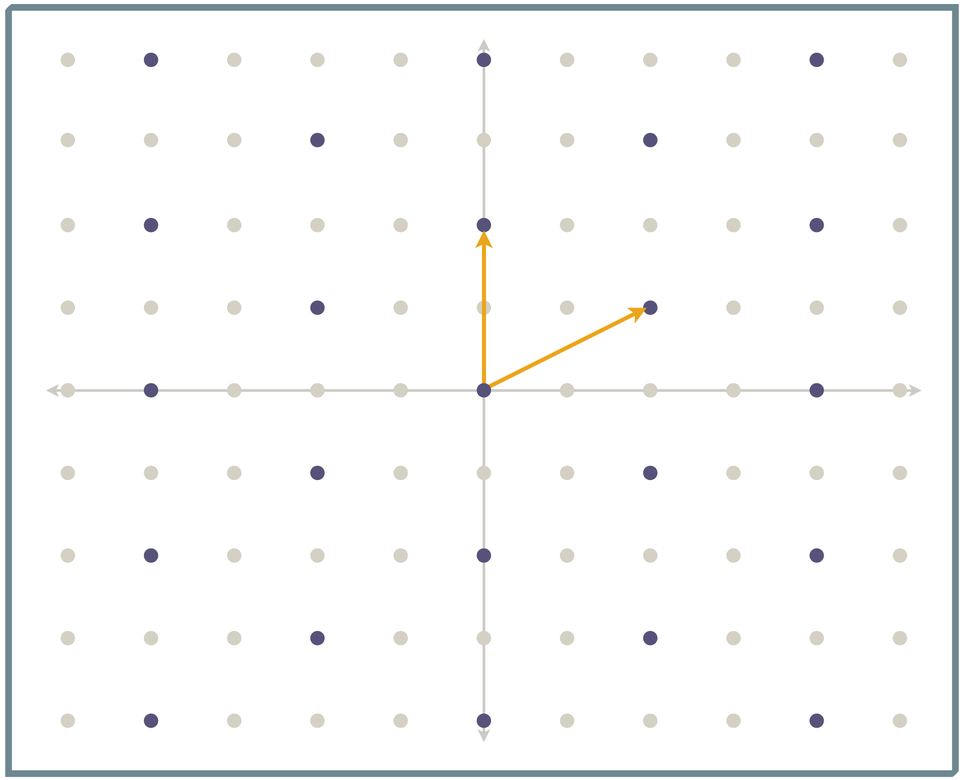
$$\left\{ A \cdot u + v \;\middle|\; u \; v \in \mathbb{Z}^n \times q\mathbb{Z}^m \right\}$$

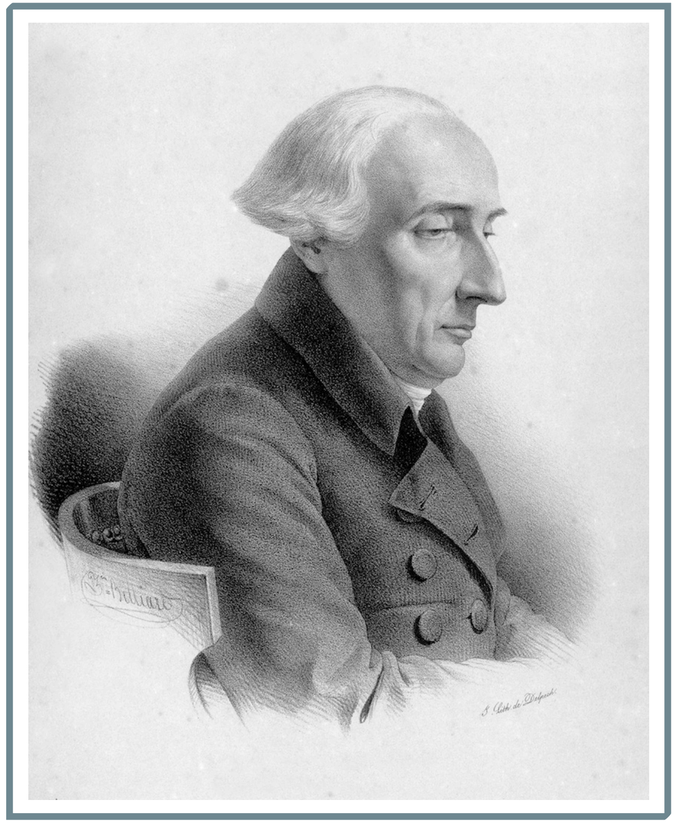
LWE

what is lattice reduction ?

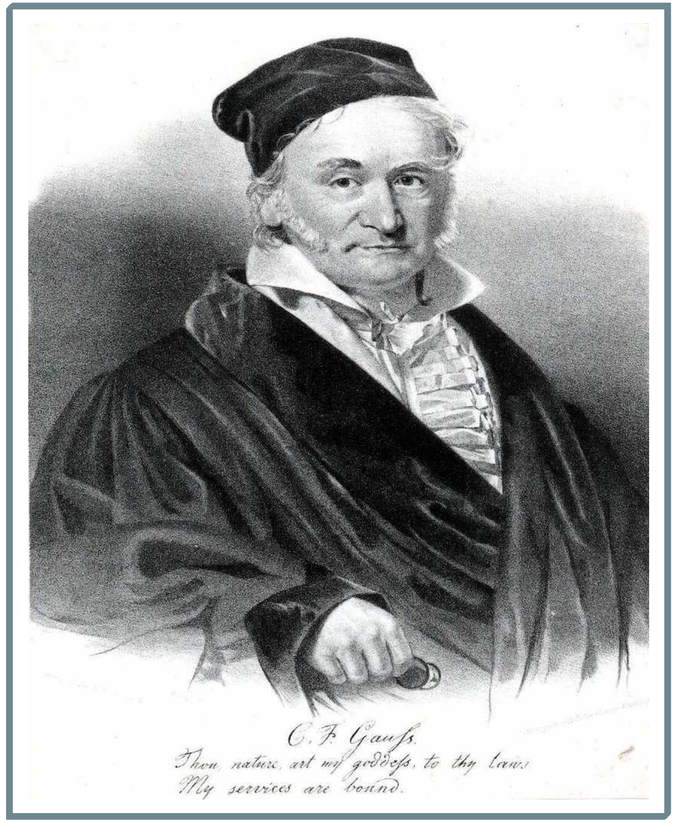




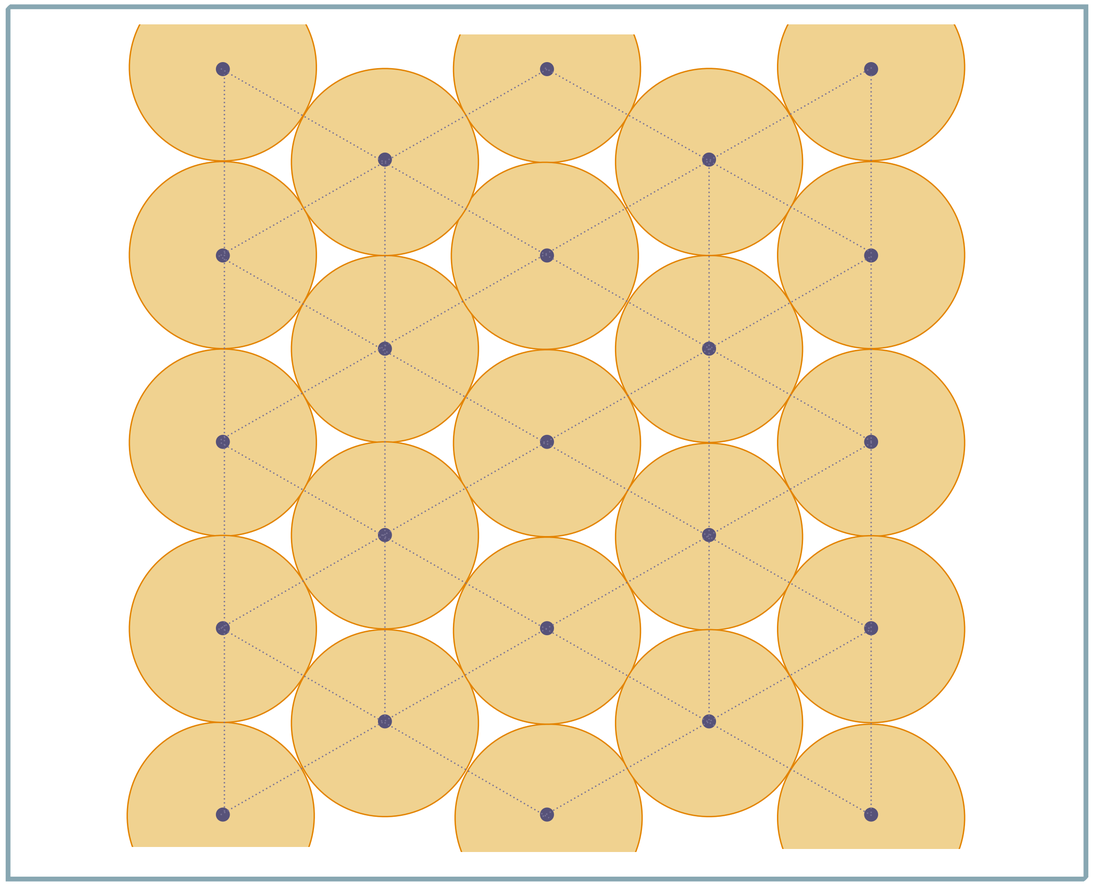




Joseph L. Lagrange

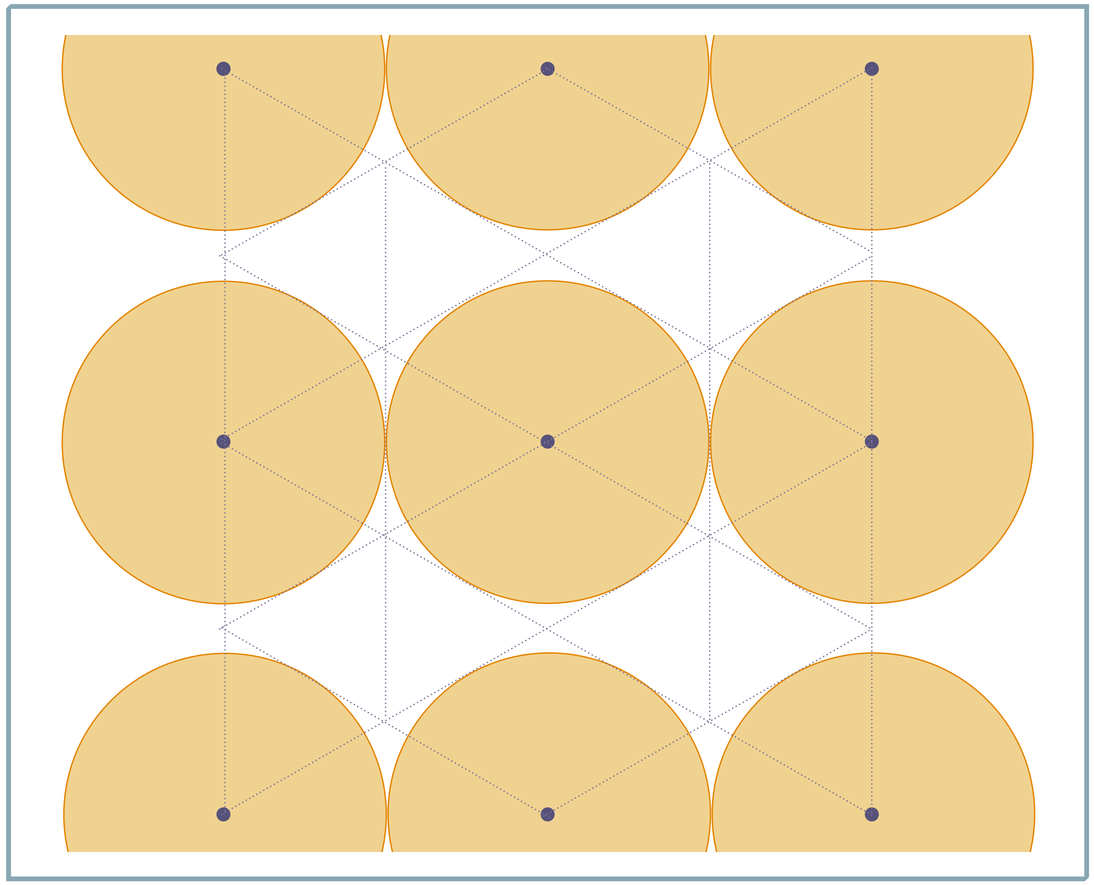


Carl F. Gauss



Minkowski theorem

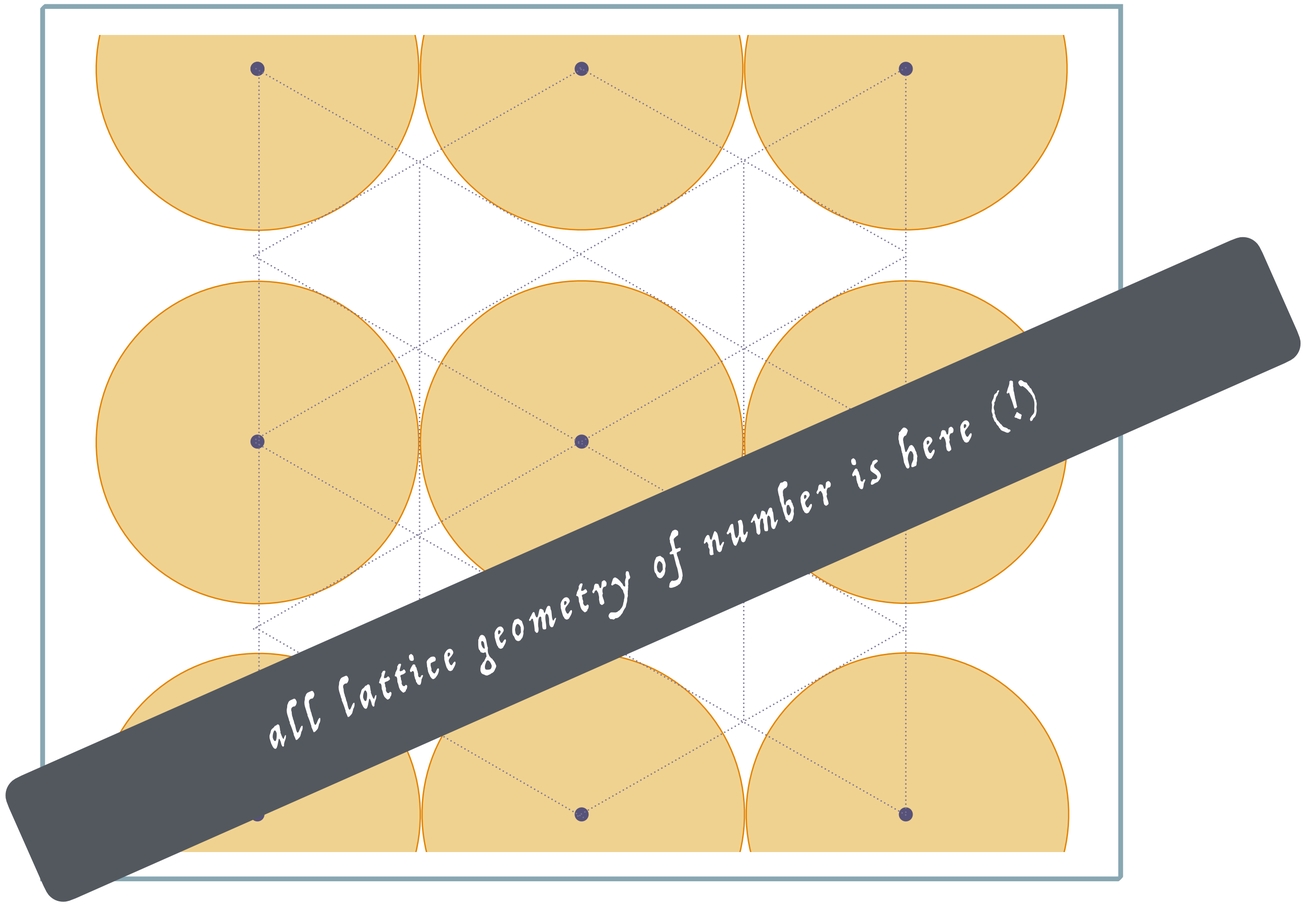
$$\text{shortest vector} \leq \sqrt{\frac{4}{3}} \frac{1}{\text{density}}$$



Minkowski theorem

$$\text{shortest vector} \leq \sqrt{\frac{4}{3}} \frac{1}{\text{density}}$$

THE (SQUARE, SCALED) LATTICE IN DIMENSION 2



Minkowski theorem

$$\text{shortest vector} \leq \sqrt{\frac{4}{3}} \frac{1}{\text{density}}$$

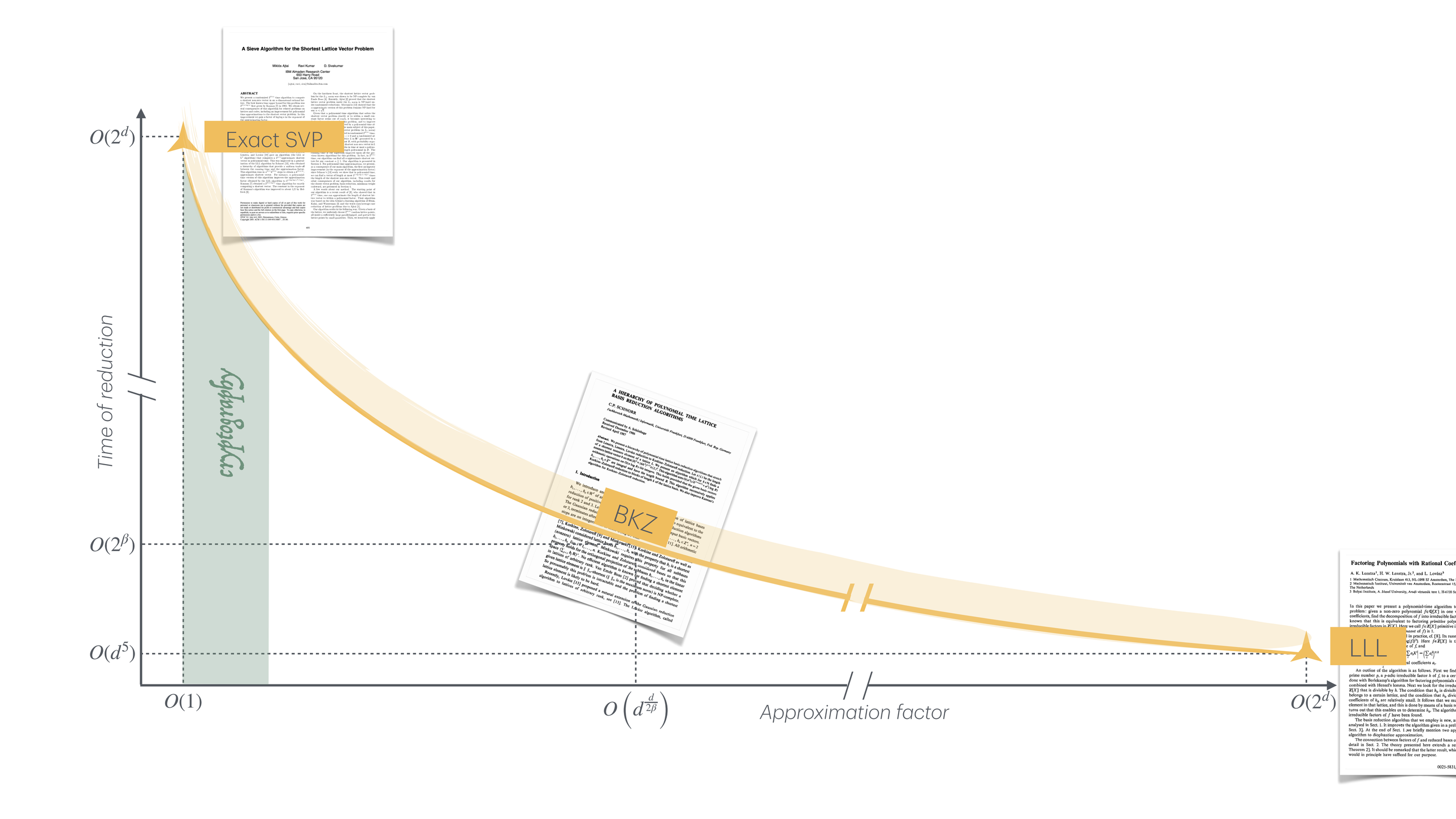
Minkowski theorem

$$\text{shortest vector} \lessapprox \sqrt{d}\frac{1}{\text{density}}$$

Shortest vectors *exist* but...	are *terribly* hard to find...	we can only *approximate*

find the shortest vector costs *exponential* time

exponentially bad approximation costs *polynomial time*


A Sieve Algorithm for the Shortest Lattice Vector Problem
Miklós Ajtai Ravi Kumar D. Sivakumar
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
{ajtai, ravi, siva}@almaden.ibm.com
ABSTRACT
Exact SVP
A HIERARCHY OF POLYNOMIAL TIME LATTICE BASIS REDUCTION ALGORITHMS
C.P. SCHNORR
BKZ
Factoring Polynomials with Rational Coef
A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász
LLL
$O(2^d)$
$O(2^\beta)$
$O(d^5)$
$O(1)$
$O\left(d^{\frac{d}{2\beta}}\right)$
$O(2^d)$
Time of reduction
Approximation factor
cryptography

that being said... too slow to run, too hard to predict

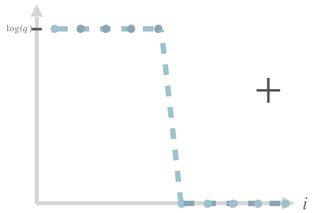
oops ! (ter)

Input lattice basis

```
[-14  -5  -5  -5 -12   3  13  -4   2   1]
[ -7   9  12   2 -15   0  13  -1  12  -2]
[-15   4  10  -2   0 -12  12   1  10   9]
[  8   6 -11 -12  14   9  13 -13   5   3]
[ -6   1  -7  14 -13  12   5  -1 -10 -15]
[ -2  11   0  -9   8   3  -8  -7  -7   0]
[  6  -2 -12   9   2 -14   6  -2  14  -8]
[  0 -12   7   9  11 -15   9  13   5   1]
[ -7   9   7 -12  10  -8  -2  11  -9  -8]
[-10   9   3 -12 -14 -13  -8 -12  11  -4]
```

simulator

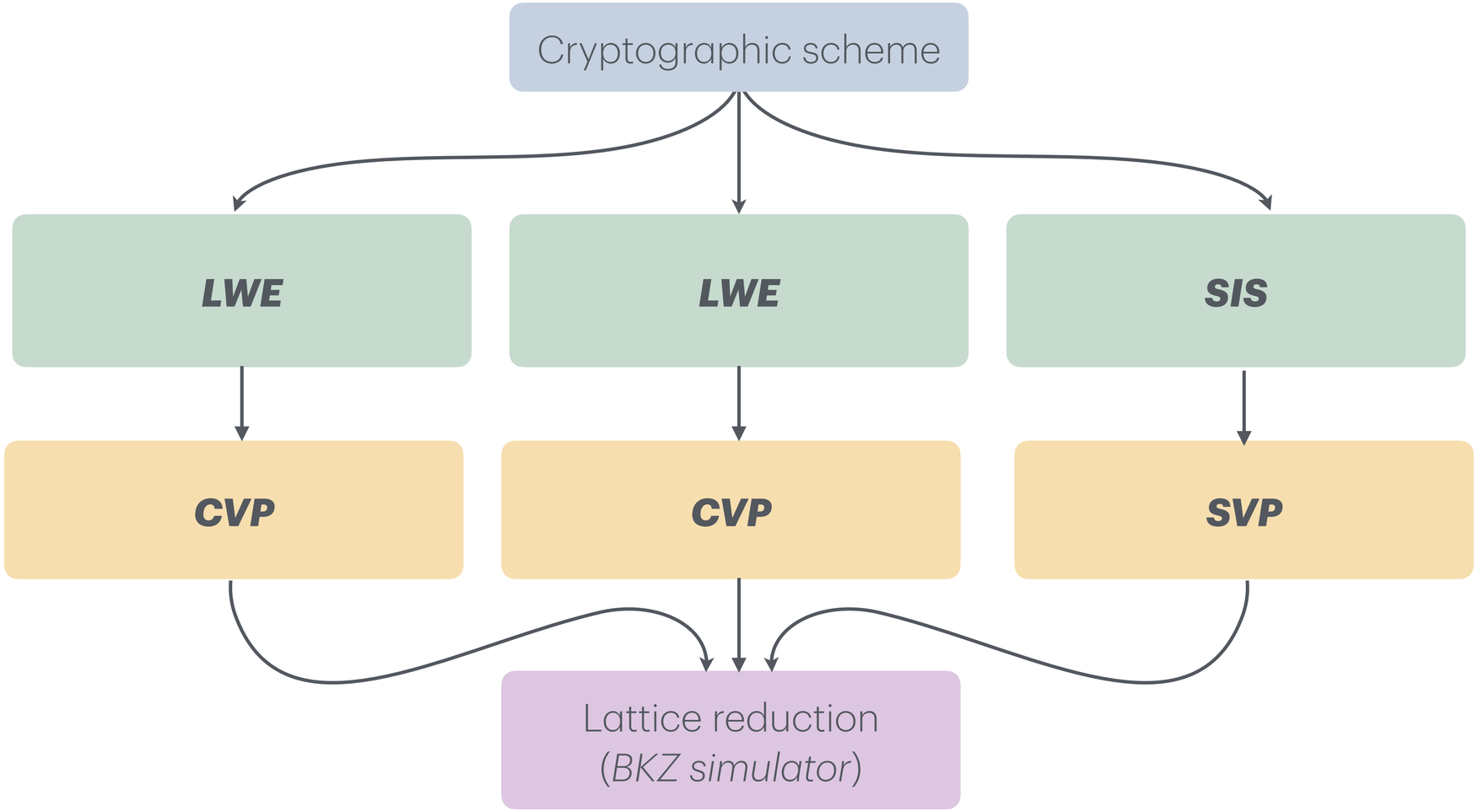
Output datum



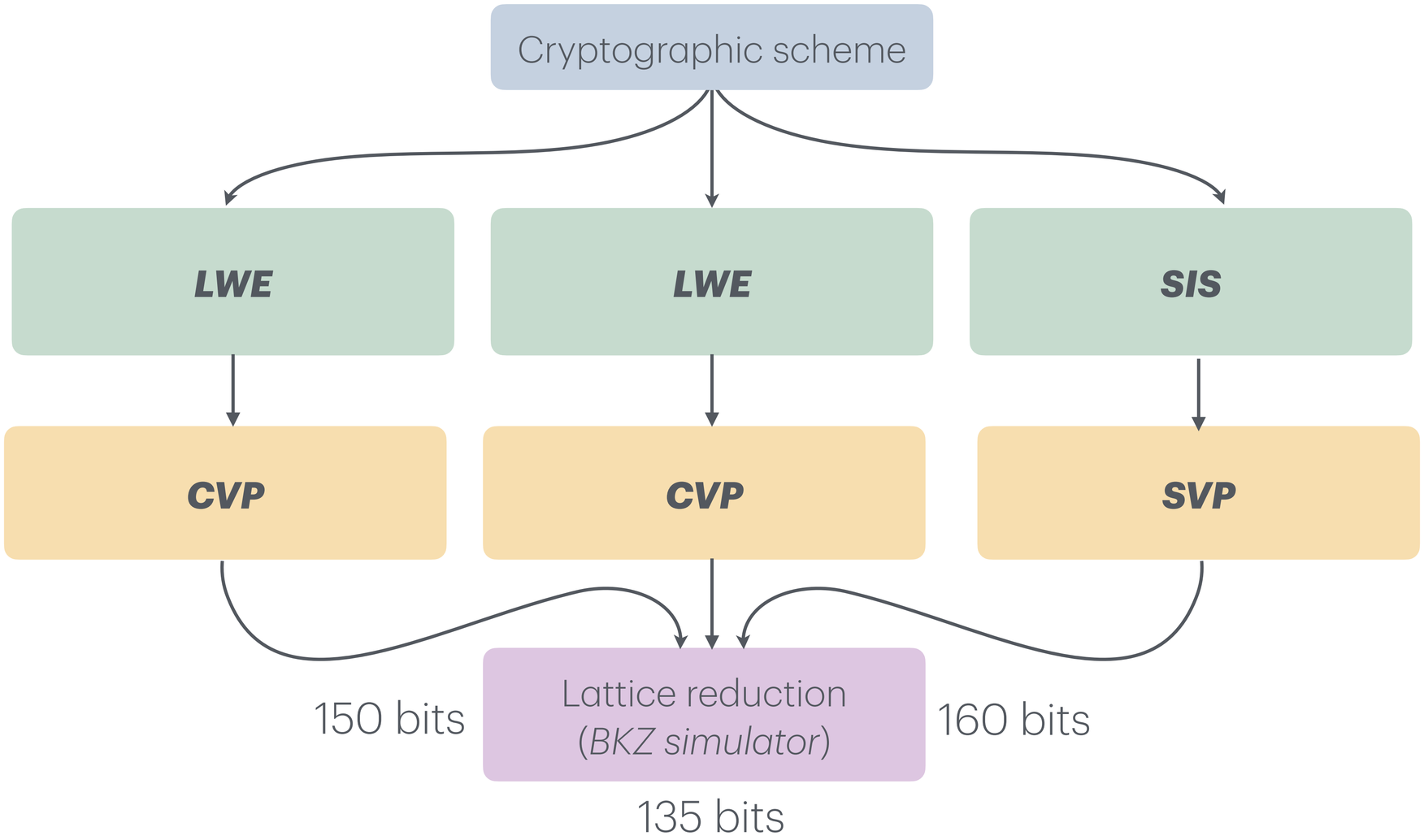
+ runtime

How do we set parameters ?

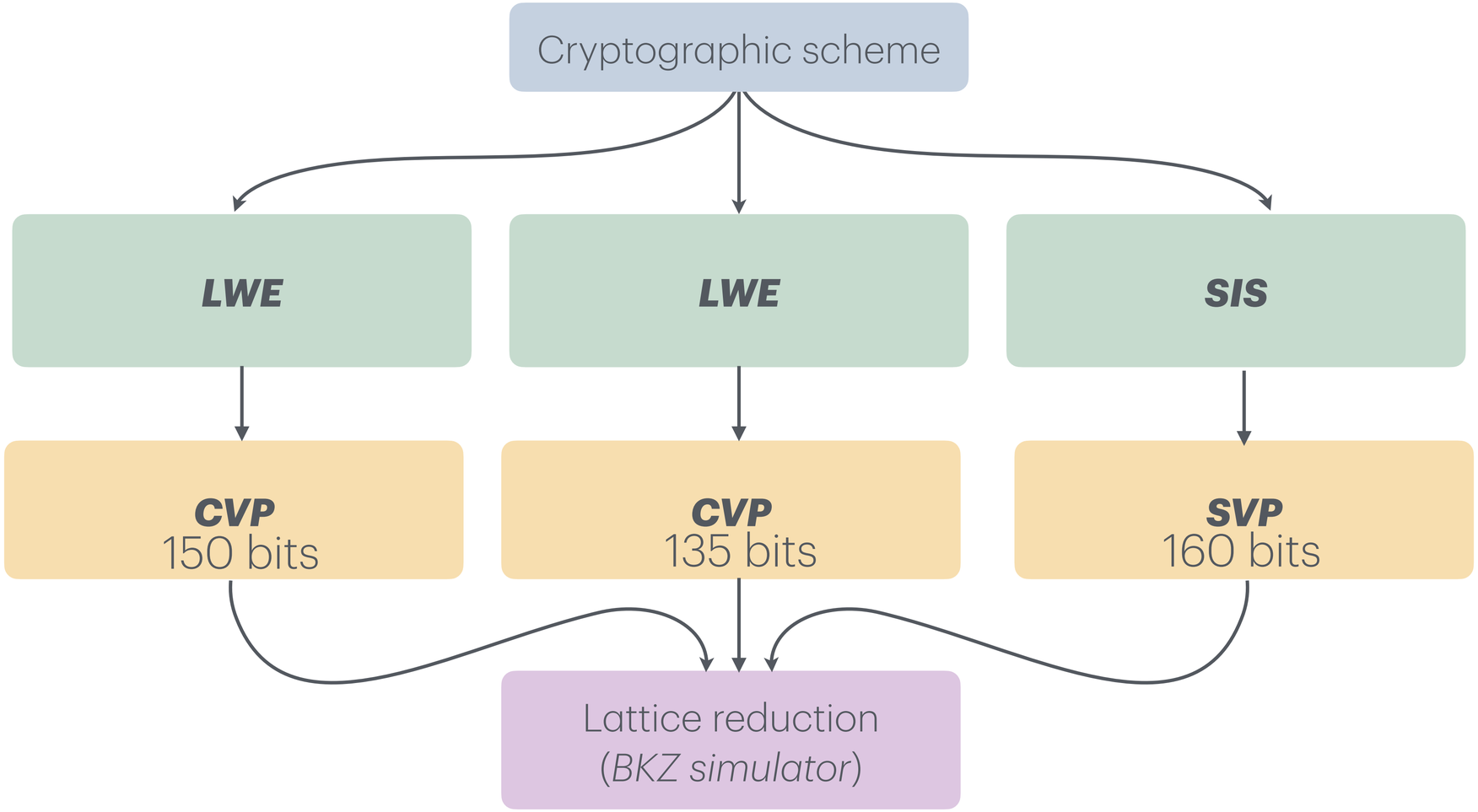
Flowchart time!



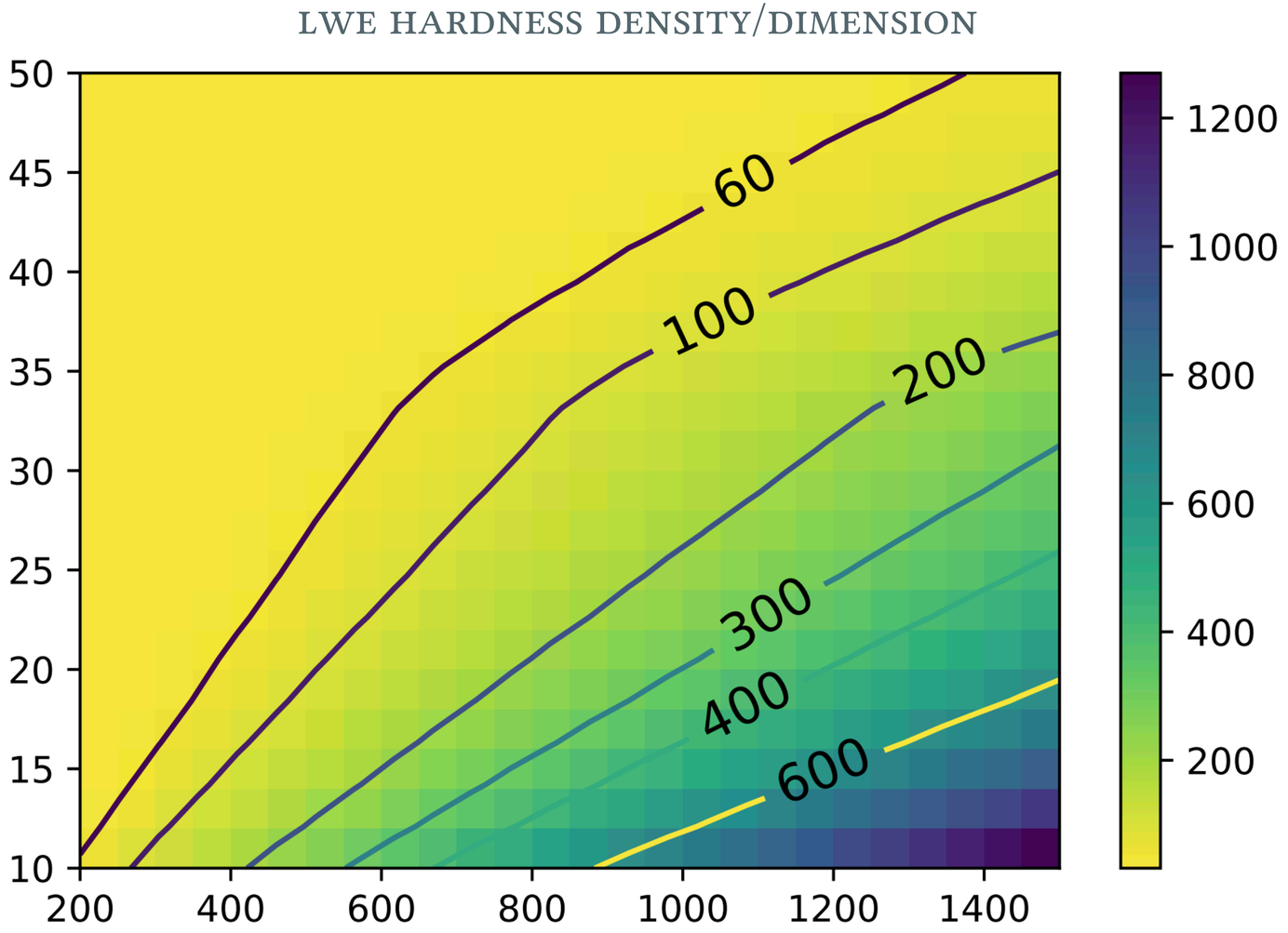
Flowchart time!



Flowchart time!




Input *parametrized* scheme
$n = 256$
$k = 5, \quad \ell = 6$
...
security estimator
Output
135 bits classical
127 bits quantum


LWE HARDNESS DENSITY/DIMENSION
50
45
40
35
30
25
20
15
10
200
400
600
800
1000
1200
1400
60
100
200
300
400
600
1200
1000
800
600
400
200

Parameter	Explanation
$\mathcal{R}_q$	Polynomial ring $\mathcal{R}_q = \mathbb{Z}[X]/(q, X^n + 1)$
(k, ℓ)	Dimension of public matrix $\mathbf{A} \in \mathcal{R}_q^{k \times \ell}$
$(\mathcal{D}_{\mathbf{t}}, \sigma_{\mathbf{t}})$	Gaussian distribution with width $\sigma_{\mathbf{t}}$ used for the verification key $\mathbf{t}$
$(\mathcal{D}_{\mathbf{w}}, \sigma_{\mathbf{w}})$	Gaussian distribution with width $\sigma_{\mathbf{w}}$ used for the commitment $\mathbf{w}$
$\nu_{\mathbf{t}}$	Amount of bit dropping performed on verification key
$\nu_{\mathbf{w}}$	Amount of bit dropping performed on (aggregated) commitment
$(q_{\nu_{\mathbf{t}}}, q_{\nu_{\mathbf{w}}})$	Rounded moduli satisfying $(q_{\nu_{\mathbf{t}}}, q_{\nu_{\mathbf{w}}}) := (\lfloor q/2^{\nu_{\mathbf{t}}} \rfloor, \lfloor q/2^{\nu_{\mathbf{w}}} \rfloor) = (\lfloor q/2^{\nu_{\mathbf{t}}} \rceil, \lfloor q/2^{\nu_{\mathbf{w}}} \rceil)$
$\mathbb{T} \subset \mathcal{R}_q$	Set of signed monomials (see Section 3)
rep	An integer s.t. $\lvert\mathbb{T}\rvert^{\mathsf{rep}-1} \geq 2^{\lambda}$
$(\mathcal{C} \subset \mathcal{R}_q, W)$	Challenge set $\{c \in \mathcal{R}_q \mid \lVert c\rVert_\infty = 1 \wedge \lVert c\rVert_1 = W\}$ s.t. $\lvert\mathcal{C}\rvert \geq 2^{\lambda}$
B	Two-norm bound on the signature

PARAMETERS OF 2 ROUND THRACCOON , CRYPTO 2024

oops ! (quad)

	q	$u_{\mathbf{t}}$	$u_{\mathbf{w}}$	$d \cdot \mathsf{rep}$	$\nu_{\mathbf{t}}$	$\nu_{\mathbf{w}}$	n	ℓ	k	ω
MLWE	↘	↗↗	=	↗	=	=	↗↗	↗↗	=	=
SelfTargetMSIS	↗↗	=	↘↘	↘	↘	↘	↗↗	↗	↗↗	↘
ϵ_{TAIL}	=	↘↘	↗↗	↗↗	=	=	↘	↘	↘	↘
$R_{\alpha}^{\epsilon_{\text{TAIL}}}$	=	↘↘	↗↗	=	=	=	↘	↘	↘	↘
Size of vk	↘	↗	=	=	↗↗	=	↘↘	=	↘↘	=
Size of sig	↘	↘	↘	↘	=	↗↗	↘↘	↘↘	↘	=

PARAMETERS OF THRACCOON , EUCROCRYPT 2024

size v. security

	q	$u_{\mathbf{t}}$	$u_{\mathbf{w}}$	$d \cdot \mathsf{rep}$	$\nu_{\mathbf{t}}$	$\nu_{\mathbf{w}}$	n	ℓ	k	ω
MLWE	↘	↗↗	=	↗	=	=	↗↗	↗↗	=	=
SelfTargetMSIS	↗↗	=	↘↘	↘	↘	↘	↗↗	↗	↗↗	↘
ϵ_{TAIL}	=	↘↘	↗↗	↗↗	=	=	↘	↘	↘	↘
$R_{\alpha}^{\epsilon_{\text{TAIL}}}$	=	↘↘	↗↗	=	=	=	↘	↘	↘	↘
Size of vk	↘	↗	=	=	↗↗	=	↘↘	=	↘↘	=
Size of sig	↘	↘	↘	↘	=	↗↗	↘↘	↘↘	↘	=

PARAMETERS OF THRACCOON , EUCROCRYPT 2024

Pipeline time!