Пост-квантовые криптосистемы на решетках и кодах

Елена Киршанова

Семинар "Индустриальная математика"
Санкт-Петербург, Россия

Представлюсь

- Диссертация на тему
  “Complexity of the Learning with Errors Problem and Memory-Efficient Lattice Sieving”,
  под руководством А. Мая.
  Рурский Университет г. Бохума, Германия 2012–2016

- Пост-док под руководством Д.Штеле.
  ENS Lyon, Франция 2016–2019

- Научный сотрудник лаборатории Мат. методы защиты информации,
  БФУ им. Канта, Калининград 2019–

Область интересов: криптография на решетках, криптанализ, алгоритмы для трудных задач на решетках и кодах.

Пост-квантовая криптография

– это классические крипто-протоколы, предположительно стойкие к атакам на квантовом компьютере.

Три основных направления пост-квантовой криптографии

I Криптография на решётках

II Криптография на кодах

III Криптография на изогениях

Кроме этого: схемы подписи на системах полиномиальных уравнений и хэш-функциях (à la подпись Лэмпорта).

План

- Сложные задачи на Евклидовы решётках
- Крипто на решетках: задача LWE
- Криптоанализ
- От решеток к кодам
- Крипто на кодах

Впереди много аббревиатур!

Часть I

Евклидовы решётки

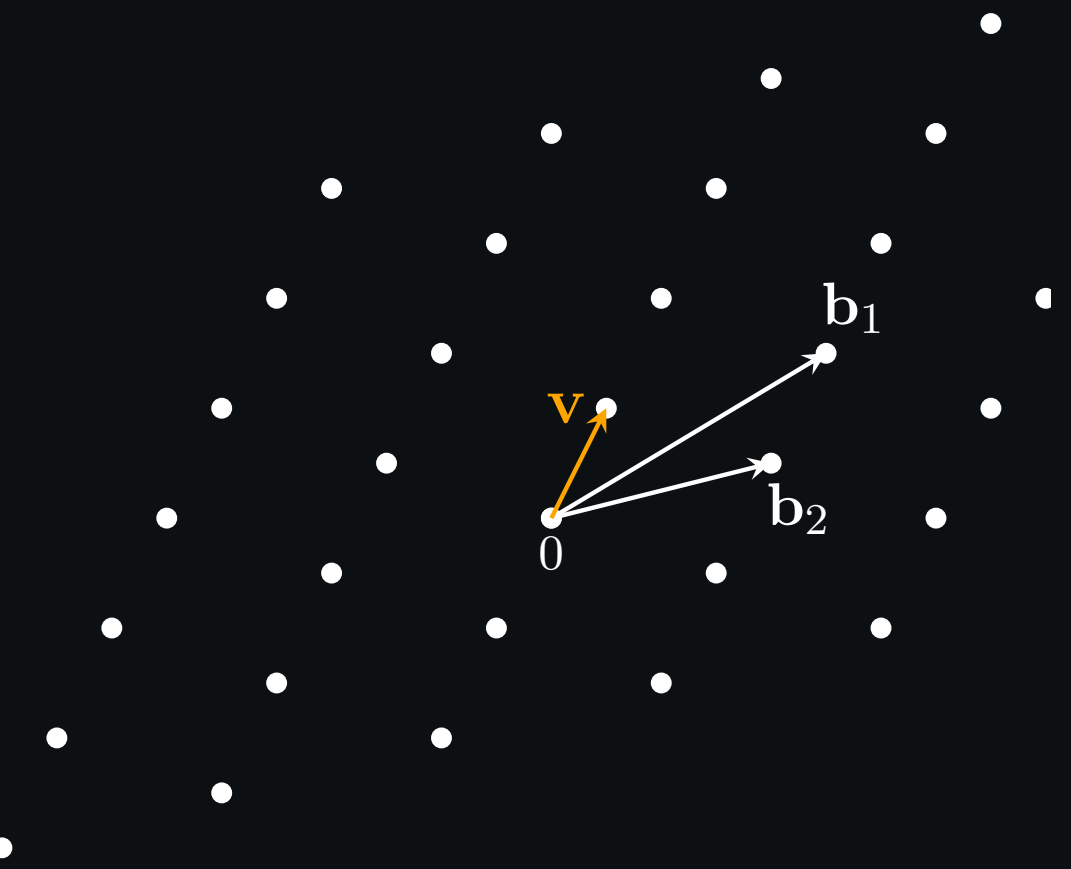
Определения

Решётка – это множество $\mathcal{L} = \{\sum_{i \le n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ для лин. независимых $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i$– базис $\mathcal{L}$, $|\{\mathbf{b}_i\}|$ – ранг $\mathcal{L}$.

Минимум

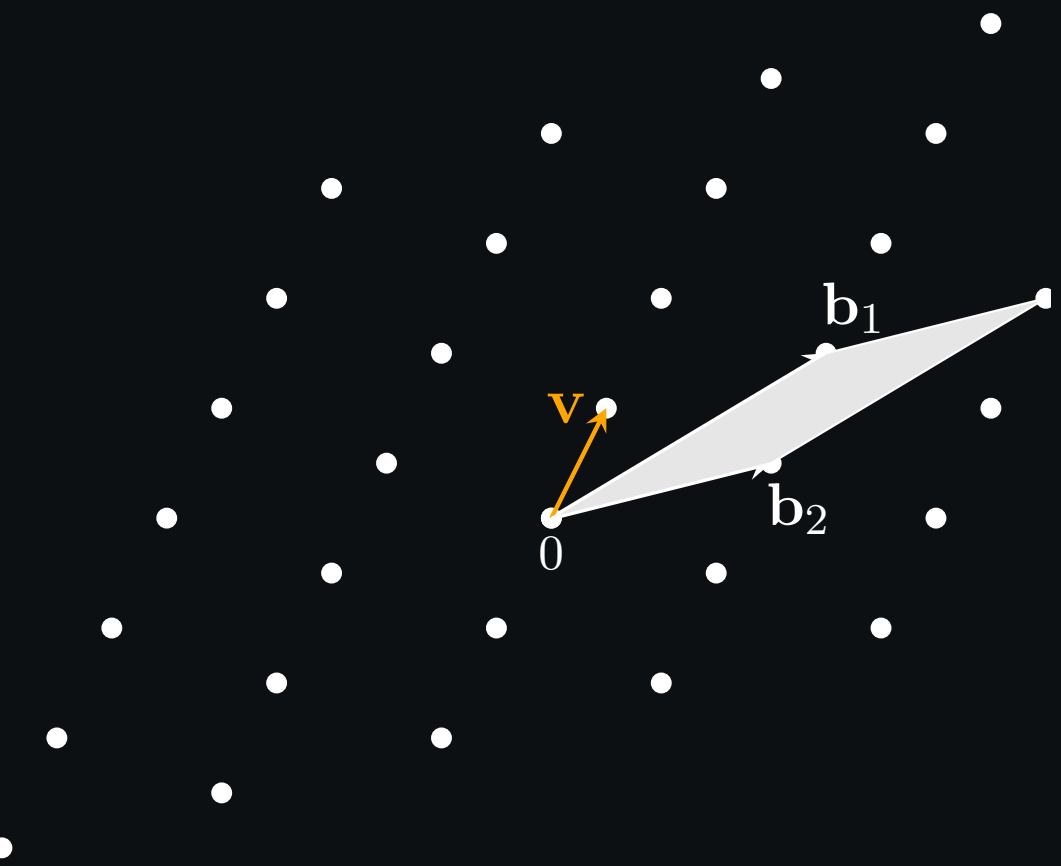
$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \mathbf{0}} \|\mathbf{v}\|$$



Решётка – это множество $\mathcal{L} = \{\sum_{i \le n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ для лин. независимых $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i$– базис $\mathcal{L}$, $|\{\mathbf{b}_i\}|$ – ранг $\mathcal{L}$.

Определения



Минимум

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \mathbf{0}} \|\mathbf{v}\|$$

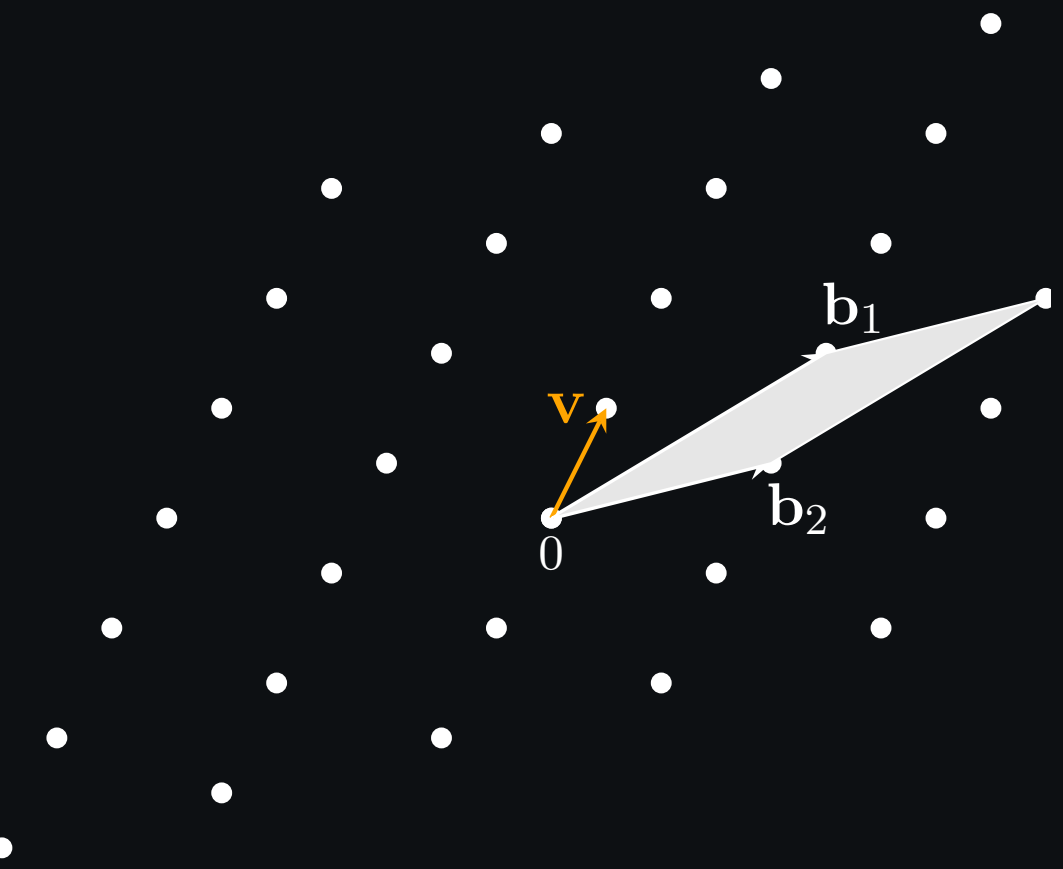
Определитель

$$\det(\mathcal{L}) = |\det(\mathbf{b}_i)_i|$$

Решётка – это множество $\mathcal{L} = \{\sum_{i \le n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ для лин. независимых $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i$– базис $\mathcal{L}$, $|\{\mathbf{b}_i\}|$ – ранг $\mathcal{L}$.

Определения



Минимум

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{v} \in \mathcal{L} \setminus \mathbf{0}} \|\mathbf{v}\|$$

Определитель

$$\det(\mathcal{L}) = |\det(\mathbf{b}_i)_i|$$

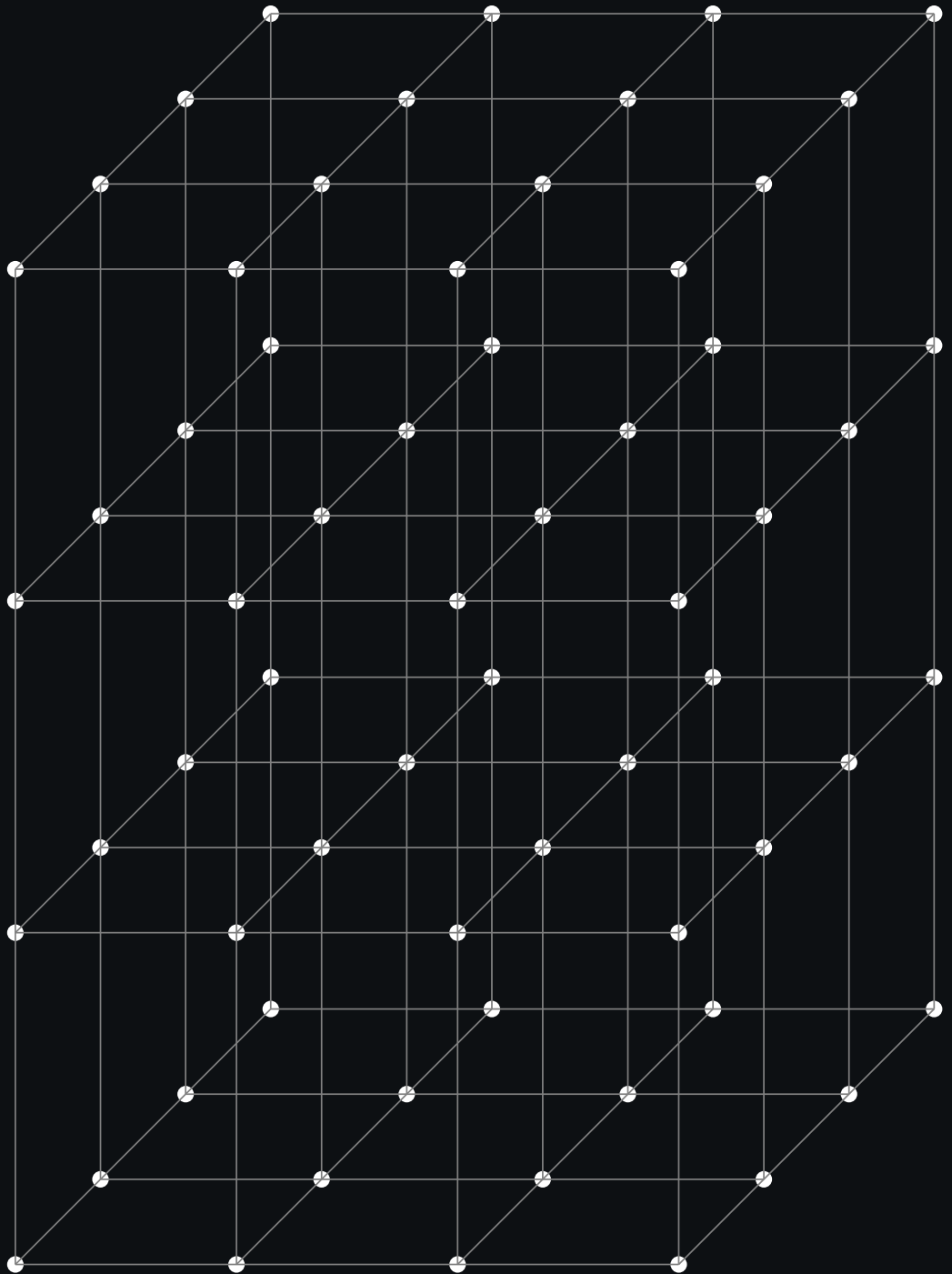
Граница Минковского

$$\lambda_1(\mathcal{L}) \leq \sqrt{n} \cdot \det(\mathcal{L})^{\frac{1}{n}}$$

Решётка – это множество $\mathcal{L} = \{\sum_{i \leq n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z}\}$ для лин. независимых $\mathbf{b}_i \in \mathbb{R}^n$

$\{\mathbf{b}_i\}_i$– базис $\mathcal{L}$, $|\{\mathbf{b}_i\}|$ – ранг $\mathcal{L}$.

Решетка $\mathbb{Z}^n$



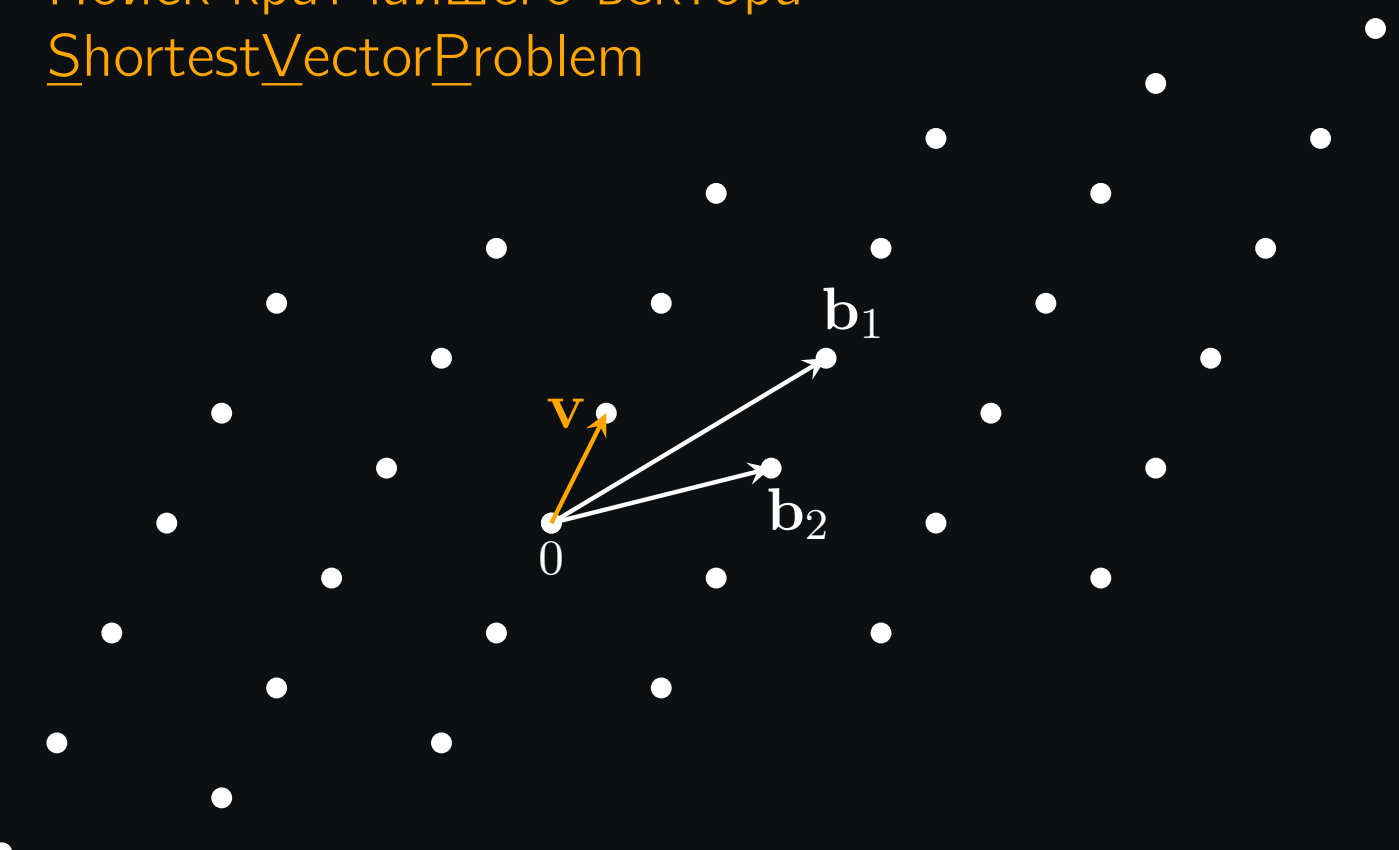
Решетка ранга n

$\det(\mathcal{L}) = 1$

$\lambda_1(\mathcal{L}) = 1$

$\mathbf{b}_i = \mathbf{e}_i.$

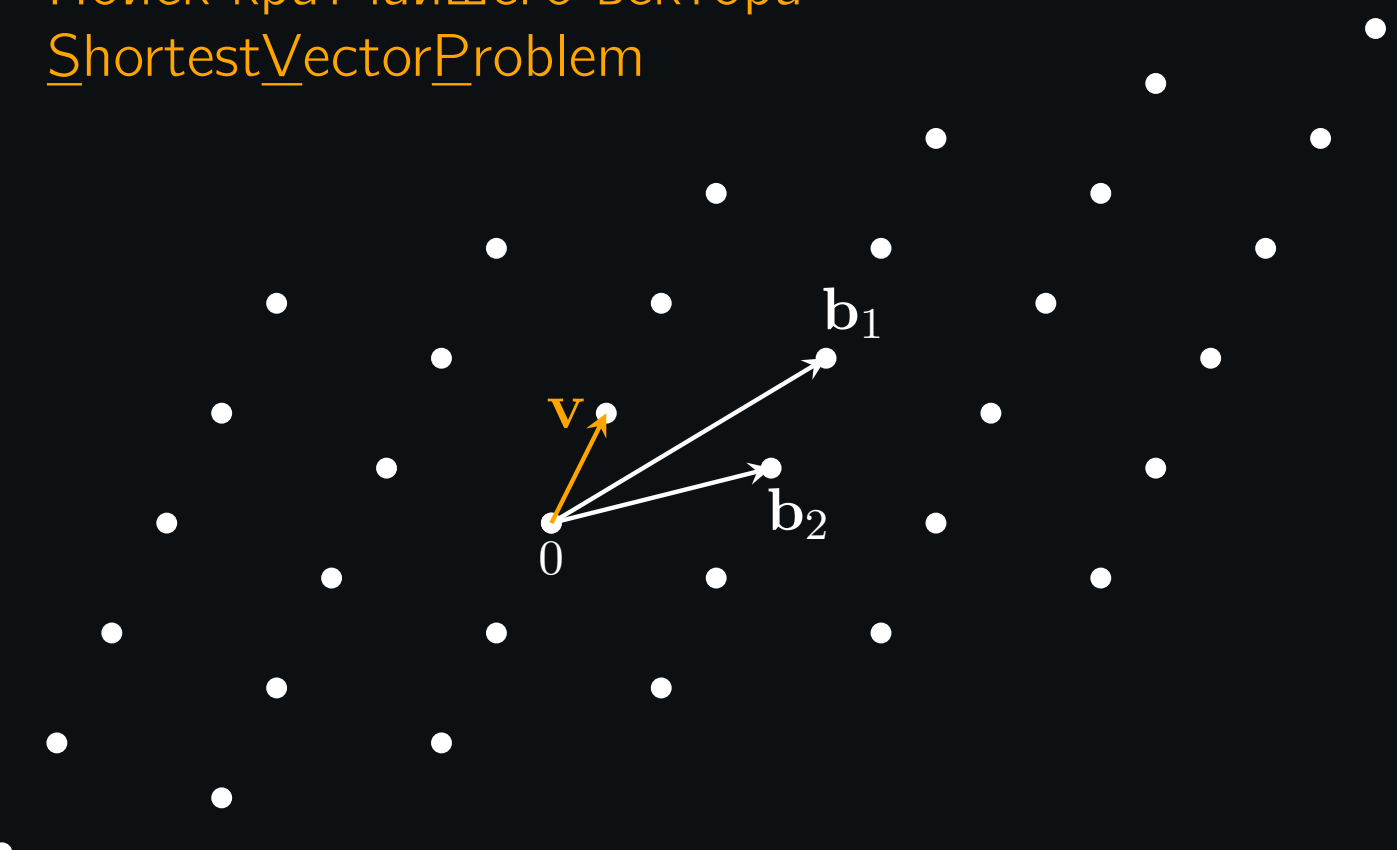
Поиск кратчайшего вектора
ShortestVectorProblem



В задаче поиска кратчайшего вектора (SVP) требуется найти $\mathbf{v}_{\text{shortest}} \in \mathcal{L}$:

$$\|\mathbf{v}_{\text{shortest}}\| = \lambda_1(\mathcal{L})$$

Поиск кратчайшего вектора
ShortestVectorProblem



В задаче поиска кратчайшего вектора (SVP) требуется найти $\mathbf{v}_{\text{shortest}} \in \mathcal{L}$:

$$\|\mathbf{v}_{\text{shortest}}\| = \lambda_1(\mathcal{L})$$

Упрощение: поиск аппроксимации (γ-SVP) к $\mathbf{v}_{\text{shortest}}$:

$$\|\mathbf{v}_{\text{short}}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$$

Поиск ближайшего вектора CVP / BDD

В задаче поиска ближайшего вектора (CVP) для $\mathbf{t} \notin \mathcal{L}$ требуется найти $\mathbf{v} \in \mathcal{L}$:

$$\|\mathbf{v} - \mathbf{t}\| \text{ минимально для всех } \mathbf{v} \in \mathcal{L}$$

Поиск ближайшего вектора CVP / BDD

В задаче поиска ближайшего вектора (CVP) для $\mathbf{t} \notin \mathcal{L}$ требуется найти $\mathbf{v} \in \mathcal{L}$:

$$\|\mathbf{v} - \mathbf{t}\| \text{ минимально для всех } \mathbf{v} \in \mathcal{L}$$

Часто: $\text{dist}(t, \mathcal{L}) \leq \frac{1}{\gamma}\lambda_1(\mathcal{L})$. Получаем задачу Декодирования с Ограниченным расстоянием, γ-BDD

Поиск ближайшего вектора CVP / BDD

Для решения BDD в $\mathcal{L}$, вызываем оракул для approx-SVP в “связанной” решетке р-ти+ 1.

В задаче поиска ближайшего вектора (CVP) для $\mathbf{t} \notin \mathcal{L}$ требуется найти $\mathbf{v} \in \mathcal{L}$:

$$\|\mathbf{v} - \mathbf{t}\| \text{ минимально для всех } \mathbf{v} \in \mathcal{L}$$

Часто: $\text{dist}(t, \mathcal{L}) \leq \frac{1}{\gamma}\lambda_1(\mathcal{L})$. Получаем задачу Декодирования с Ограниченным расстоянием, γ-BDD

От задачи BDD к approxSVP: вложение Каннана

Для задачи BDD $(\mathcal{L}, \mathbf{t})$, где $\mathcal{L}$ имеет базиз B, рассмотрим для константы c

$$B' = \begin{bmatrix} B & \mathbf{t} \\ \mathbf{0} & \mathrm{c} \end{bmatrix}$$

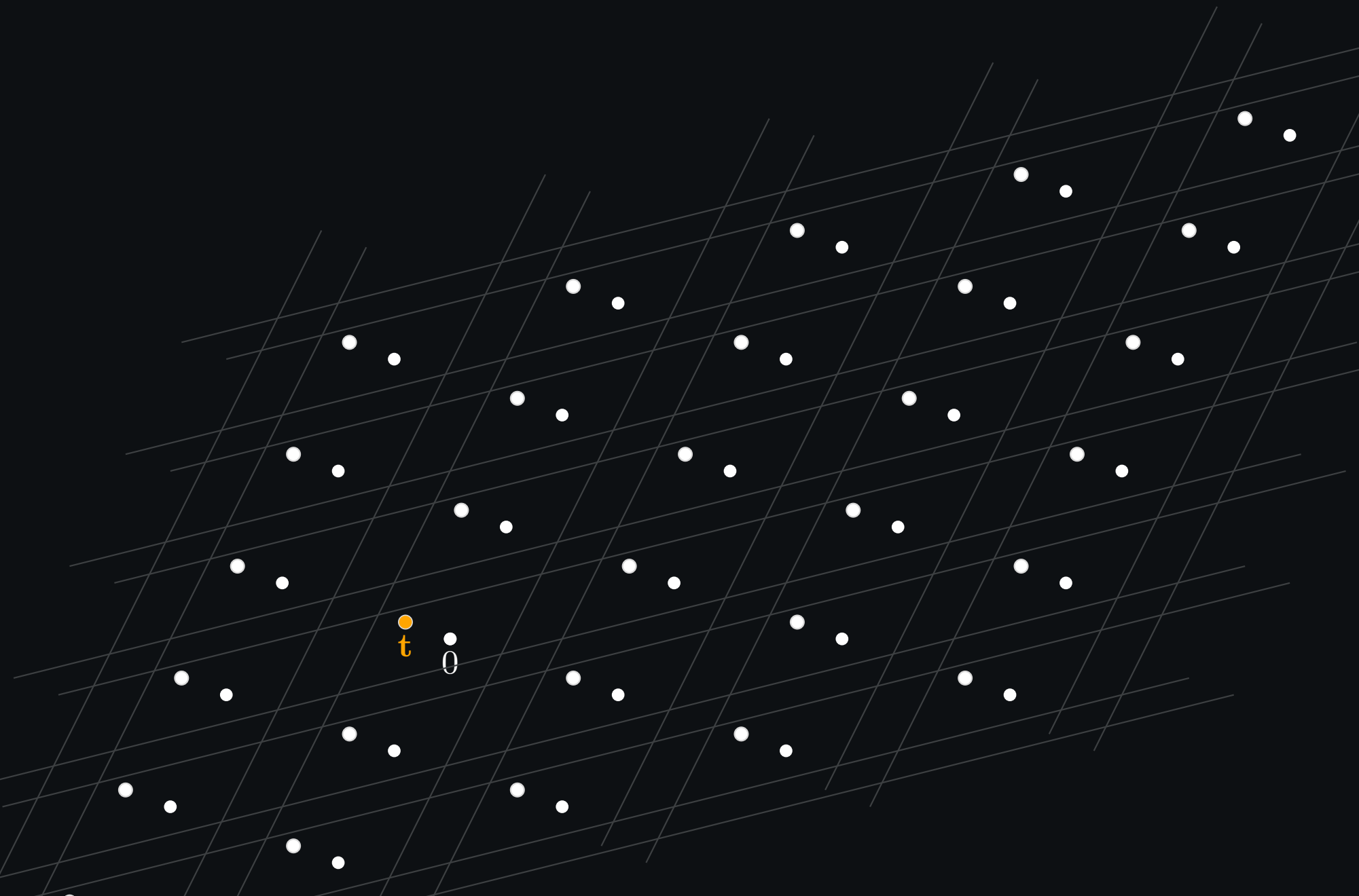
- столбцы B' лин. независимы
- Для "грамотно" выбранной c и $\mathbf{t}$ - достаточно близкого к $\mathcal{L}$,

$$\begin{bmatrix} B & \mathbf{t} \\ \mathbf{0} & \mathrm{c} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \begin{bmatrix} B\mathbf{x} - \mathbf{t} \\ \mathrm{c} \end{bmatrix}$$

– короткий вектора в $\mathcal{L}(B')$ (намного короче, чем любой $\mathbf{v} \in \mathcal{L}(B')$ не параллельный ему).

От задачи BDD к approxSVP: вложение Каннана

От задачи BDD к approxSVP: вложение Каннана

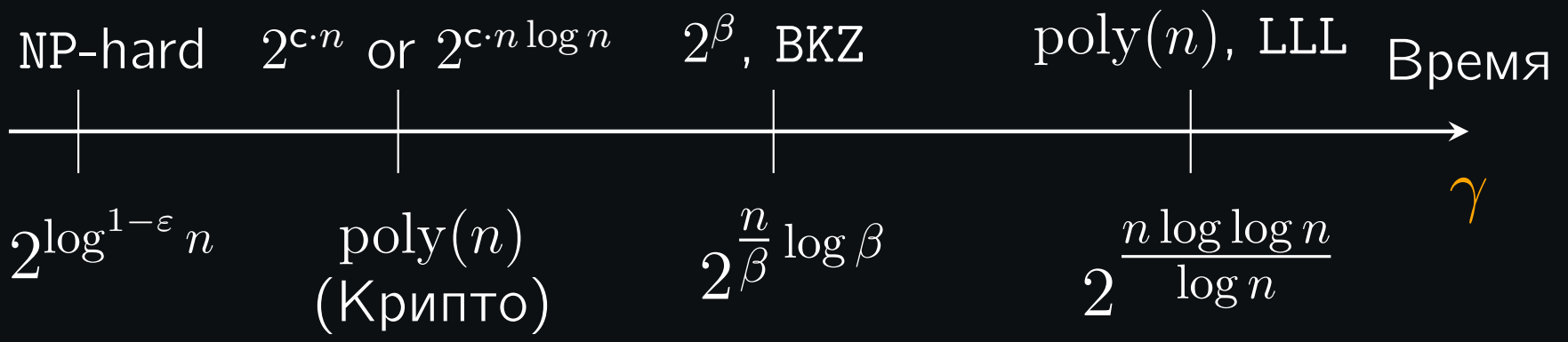


Асимптотическая сложность SVP

$$||\mathbf{v}_{\text{short}}|| \leq \gamma \cdot ||\mathbf{v}_{\text{shortest}}||$$

Асимптотическая сложность SVP

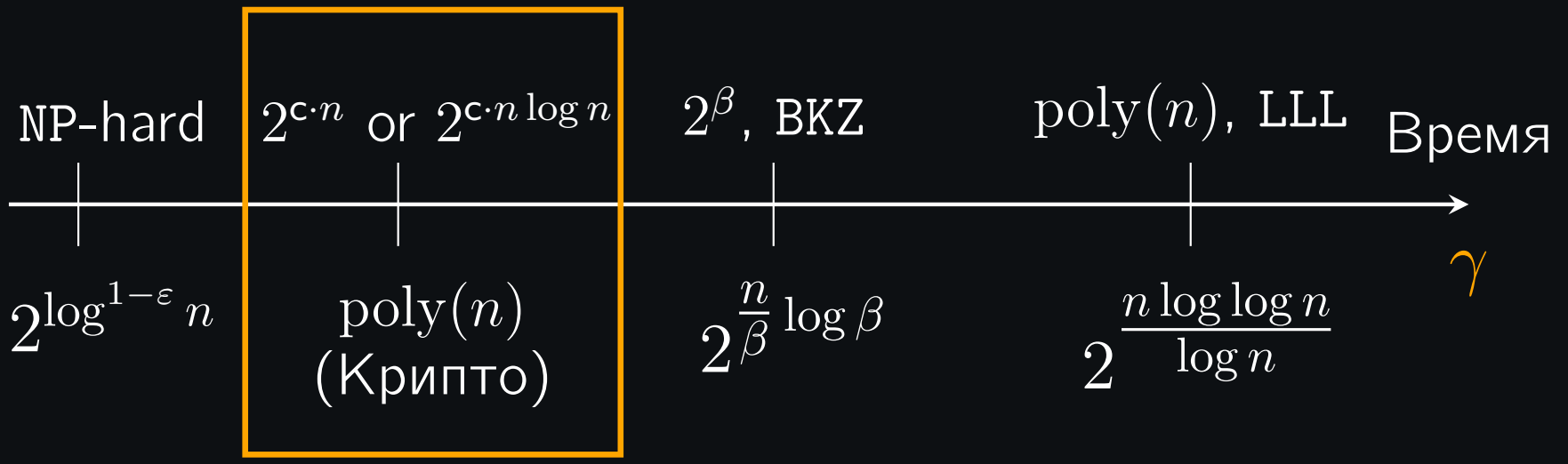
$$||\mathbf{v}_{\text{short}}|| \leq \gamma \cdot ||\mathbf{v}_{\text{shortest}}||$$



BKZ – Block Korkine-Zolotarev, LLL – Lenstra, Lenstra, Lovász

Асимптотическая сложность SVP

$$||\mathbf{v}_{\text{short}}|| \leq \gamma \cdot ||\mathbf{v}_{\text{shortest}}||$$



- Просеивание (эвристический):

$$\text{Время(SVP)} = 2^{n+o(n)} \qquad \text{Память} = 2^{n+o(n)}$$

- Перечисление:

$$\text{Время(SVP)} = 2^{n \log n + o(n \log n)} \qquad \text{Память} = \text{poly}(n)$$

На сегодняшний день не существует эффективного алгоритма (ни классического, ни квантового) для задачи SVP!

BKZ – Block Korkine-Zolotarev, LLL – Lenstra, Lenstra, Lovász

Часть II

Задача обучения с ошибками (The Learning with Errors problem, LWE)

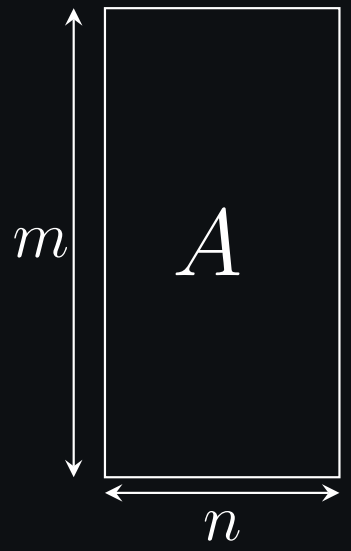
Построение решеток

Зафиксируем модуль q. $\mathbb{Z}_q$ – кольцо вычетов по модулю q.

- Выберем случайным образом $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, $m > n$
- A задает решетку

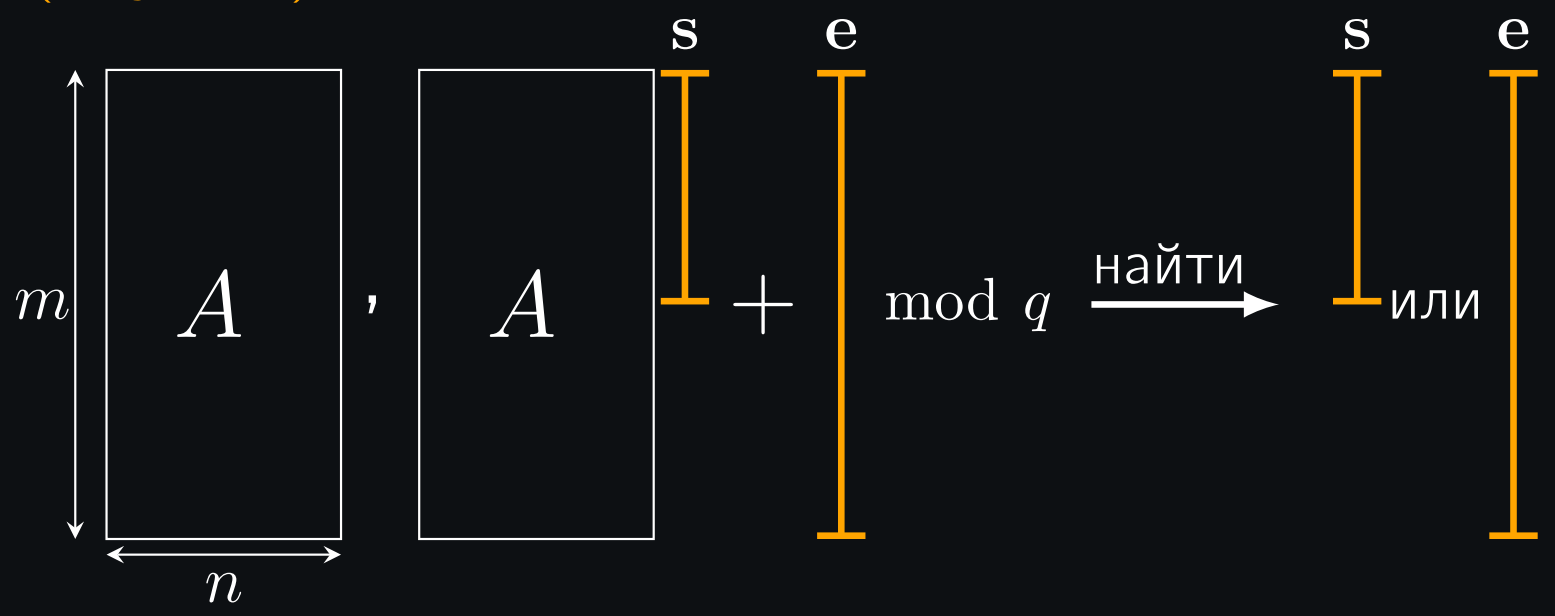
$$\mathcal{L}_q(A) = A\mathbb{Z}_q^n + q\mathbb{Z}^m$$

- С большой вероятностью $\mathcal{L}_q(A)$ ранга m и $\det(\mathcal{L}_q(A)) = q^{m-n}$



Такая конструкция носит название Конструкция A.

LWE (Regev'05)



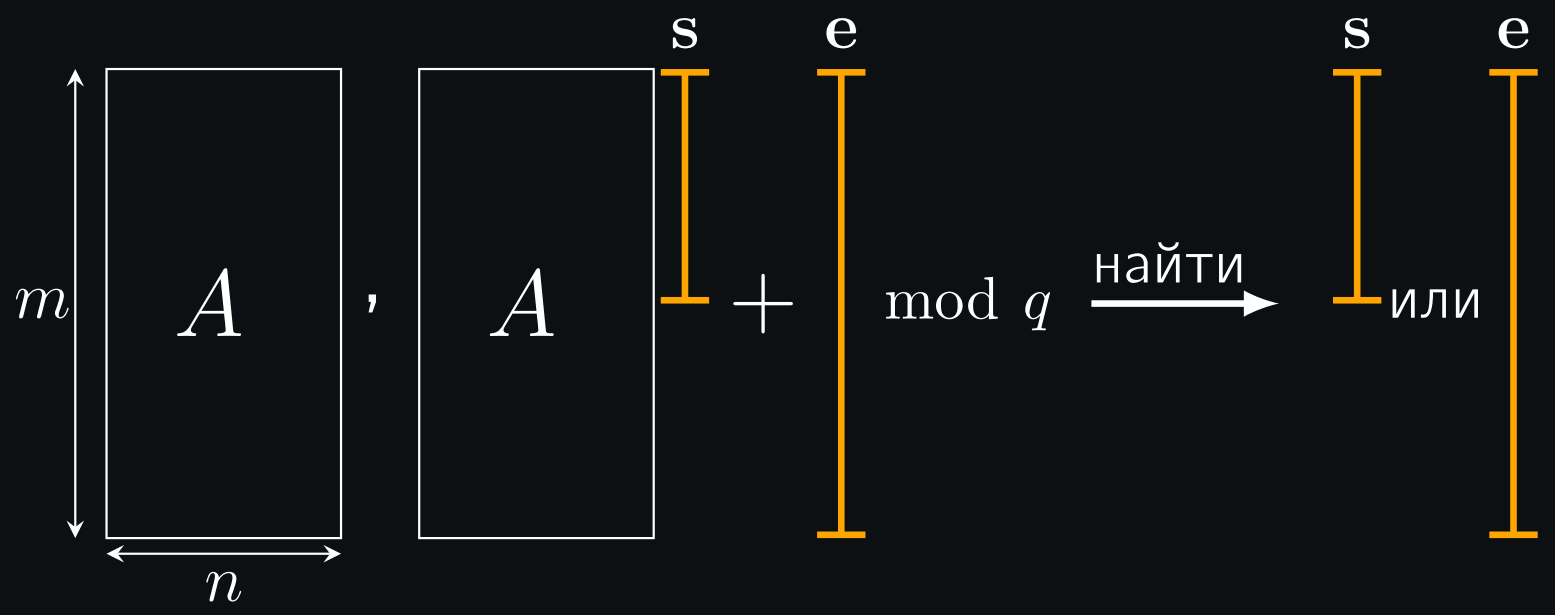
$$A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$$

$$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$$

$$\mathbf{e} \leftarrow \mathbb{Z}^m, \|\mathbf{e}\| < B$$

Часто: $n = \Theta(\text{bit security})$, $q = n^{\Theta(1)}$, $m = \Theta(n \log q)$, $B = \Omega(\sqrt{n})$ $(\sigma = \Omega(\sqrt{n}))$

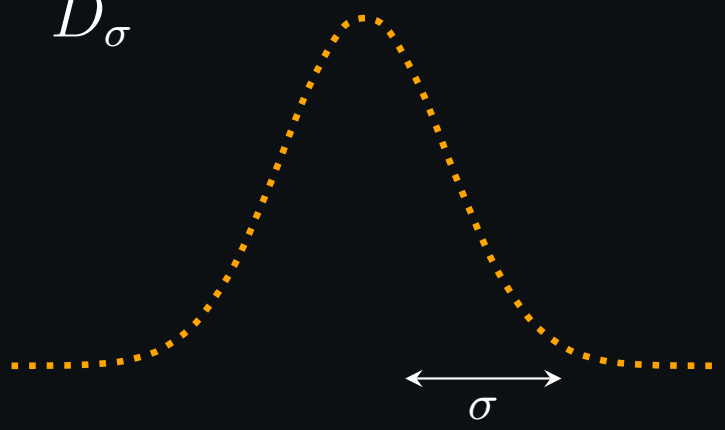
LWE (Regev'05)



$$A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$$

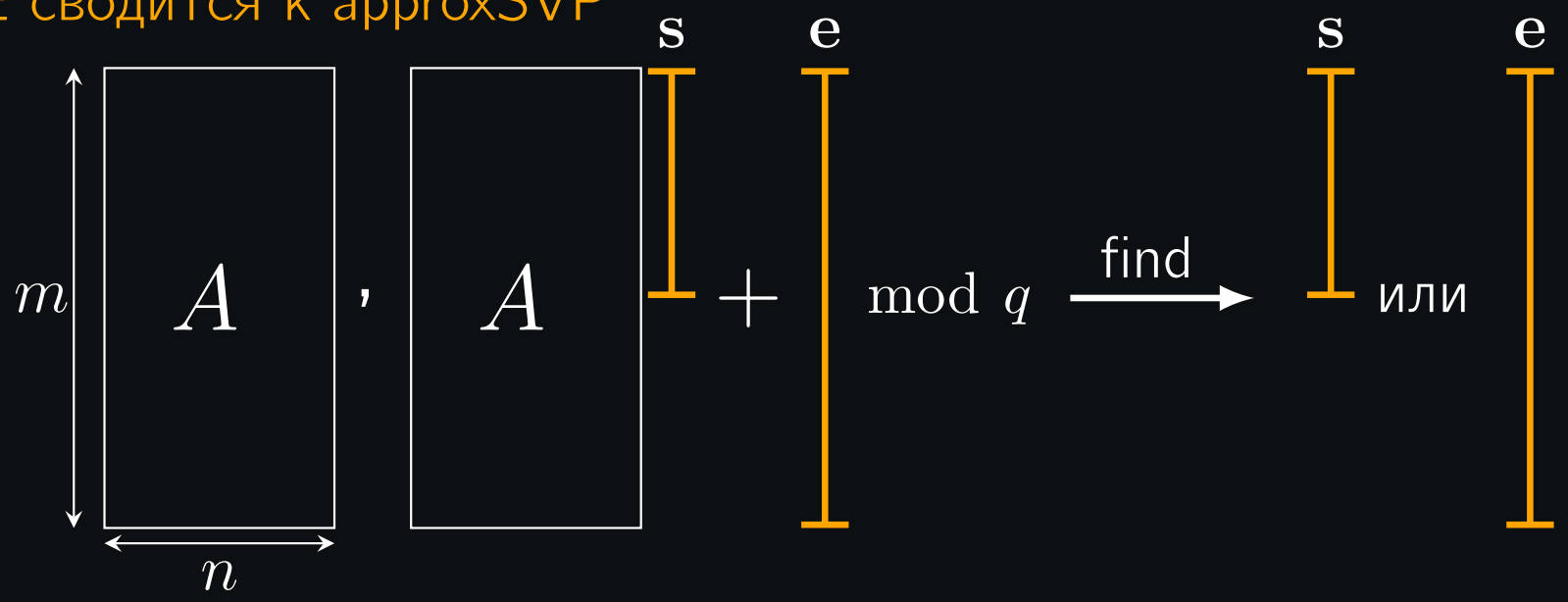
$$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^{n}$$

$$\mathbf{e} \leftarrow D_\sigma$$



Часто: $n = \Theta(\text{bit security})$, $q = n^{\Theta(1)}$, $m = \Theta(n \log q)$,
$B = \Omega(\sqrt{n})$ $(\sigma = \Omega(\sqrt{n}))$

LWE сводится к approxSVP



- A задает решётку A-Конструкции (Construction-A)

$$\mathcal{L}_q(A) = A\mathbb{Z}_q^n + q\mathbb{Z}^m$$

- $As \in \mathcal{L}_q(A)$
- $As + e$ – вектор, на расстоянии $\|e\|$ от $\mathcal{L}_q(A)$
- $(A, As + e)$ – пример BDD задачи для $\mathcal{L}_q(A)$ с

$$\gamma = \frac{\lambda_1(\mathcal{L}_q(A))}{\mathrm{dist}(As + e, \mathcal{L}_q(A))} = \frac{q^{1-n/m}}{|e_i|}$$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  - Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  - ${c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}$
  - $c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathtt{mes}$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  – Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  – ${c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}$
  – $c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathtt{mes}$

- **Dec**$(\mathsf{sk}, c = (c_1, c_2))$:
  ${c_1}^{\mathsf{t}} s - c_2 =$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  – Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  – ${c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}$
  – $c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathtt{mes}$

- **Dec**$(\mathsf{sk}, c = (c_1, c_2))$:
  ${c_1}^{\mathsf{t}} s - c_2 = u^{\mathsf{t}} A s + w^{\mathsf{t}} s - u^{\mathsf{t}} A s - u^{\mathsf{t}} e + e' + \left\lfloor \frac{q}{2} \right\rceil \mathtt{mes} =$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  – Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  – ${c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}$
  – $c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathtt{mes}$

- **Dec**$(\mathsf{sk}, c = (c_1, c_2))$:
  $${c_1}^{\mathsf{t}} s - c_2 = u^{\mathsf{t}} A s + w^{\mathsf{t}} s - u^{\mathsf{t}} A s - u^{\mathsf{t}} e + e' + \left\lfloor \frac{q}{2} \right\rceil \mathtt{mes} =$$
  $$\underbrace{w^{\mathsf{t}} s - u^{\mathsf{t}} e + e'}_{\text{шум}} + \left\lfloor \frac{q}{2} \right\rceil \mathtt{mes}$$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, b = As + e)$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  – Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  – ${c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}$
  – $c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathtt{mes}$

- **Dec**$(\mathsf{sk}, c = (c_1, c_2))$:
  $${c_1}^{\mathsf{t}} s - c_2 = u^{\mathsf{t}} A s + w^{\mathsf{t}} s - u^{\mathsf{t}} A s - u^{\mathsf{t}} e + e' + \left\lfloor \frac{q}{2} \right\rceil \mathtt{mes} =$$
  $$\underbrace{w^{\mathsf{t}} s - u^{\mathsf{t}} e + e'}_{\text{шум}} + \left\lfloor \frac{q}{2} \right\rceil \mathtt{mes}$$
  $${c_1}^{\mathsf{t}} s - c_2 \overset{?}{\approx} q/2 : \mathtt{mes} = 1, \mathtt{mes} = 0$$

Метод шифрования, основанный на LWE

Все операции в $\mathbb{Z}_q$. x – вектор-столбец, x^{t} – вектор-строка

- **KeyGen** : $\mathsf{pk} = (A, \boxed{b = As + e})$ – открытый ключ
  $\mathsf{sk} = s$ – секретный ключ

- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\})$:
  – Выберем короткие $u \in \mathbb{Z}^n, w \in \mathbb{Z}^m$ и малое $e' \in \mathbb{Z}$
  – $\boxed{{c_1}^{\mathsf{t}} = u^{\mathsf{t}} A + w^{\mathsf{t}}}$
  – $\boxed{c_2 = u^{\mathsf{t}} b + e' + \left\lfloor \frac{q}{2} \right\rceil} \cdot \mathtt{mes}$

Для злоумышленника все элементы $\boxed{}$
выглядят случайными векторами из $\mathbb{Z}_q$ под предположением сложности LWE.
Формально: сложность LWE $\implies$ CPA-безопасность схемы

LWE шифрование на практике

- Для эффективных систем (скорость операции умножения, размеры ключей) используется арифметика над фактор-кольцом $\mathbb{Z}_q[x]/(f(x))$ для неприводимого $f(x)$ большой степени
- Например, кандидат на стандартизацию NIST[1] NewHope[2] использует

$$f = x^{1024} + 1,\ q = 12289.$$

Для вектора ошибок используется биномиальное распределение с малой дисперсией и мат. ожиданием 0.

[1] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

[2] https://newhopecrypto.org/resources.shtml

LWE шифрование на практике

- Для эффективных систем (скорость операции умножения, размеры ключей) используется арифметика над фактор-кольцом $\mathbb{Z}_q[x]/(f(x))$ для неприводимого $f(x)$ большой степени
- Например, кандидат на стандартизацию NIST[1] NewHope[2] использует

$$f = x^{1024} + 1,\ q = 12289.$$

  Для вектора ошибок используется биномиальное распределение с малой дисперсией и мат. ожиданием 0.
- Существуют цифровые подписи на решетках: Dilithium, Falcon.
- Большинство примитивов на решетках значительно быстрее конструкций на кодах или изогениях.

---

[1] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization

[2] https://newhopecrypto.org/resources.shtml

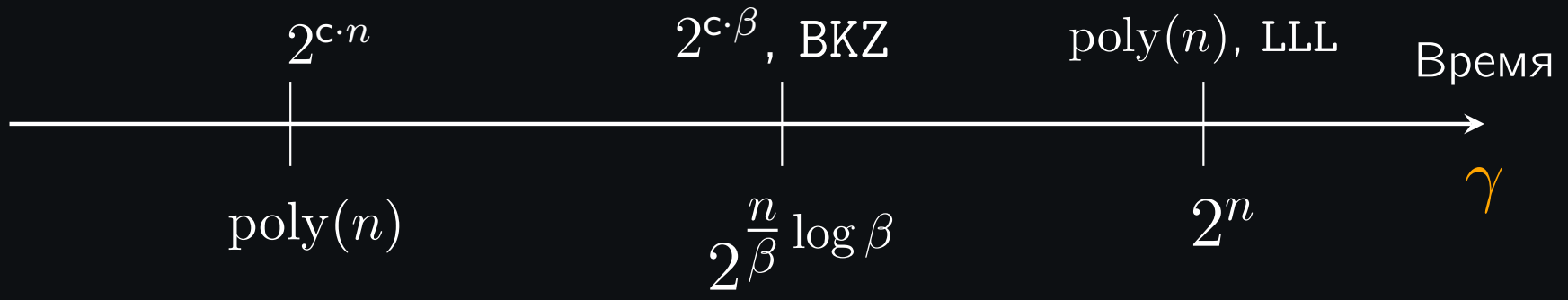
Часть III

Криптоанализ систем на решетках

Сложность LWE

LWE $(A, As + e)$ – BDD в $\mathcal{L}_q(A)$ с параметром

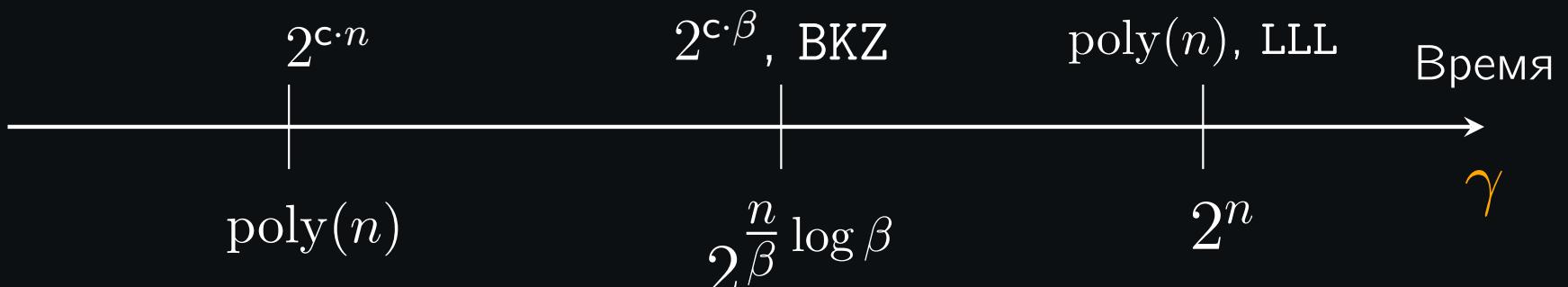
$$\gamma = \frac{\lambda_1(\mathcal{L}_q(A))}{\mathsf{dist}(As + e, \mathcal{L}_q(A))} = \frac{q^{1-n/m}}{|e_i|}$$



Сложность LWE

LWE $(A, As + e)$ – BDD в $\mathcal{L}_q(A)$ с параметром

$$\gamma = \frac{\lambda_1(\mathcal{L}_q(A))}{\mathrm{dist}(As + e, \mathcal{L}_q(A))} = \frac{q^{1-n/m}}{|e_i|}$$



В асимптотике для константы c:

$$T(\mathrm{LWE}) = \exp\left(c \cdot \frac{\lg q}{\lg^2(q/|e_i|)} \lg\left(\frac{n \lg q}{\lg^2(q/|e_i|)}\right) \cdot n\right)$$

Эта формула получена решением уравнения для β

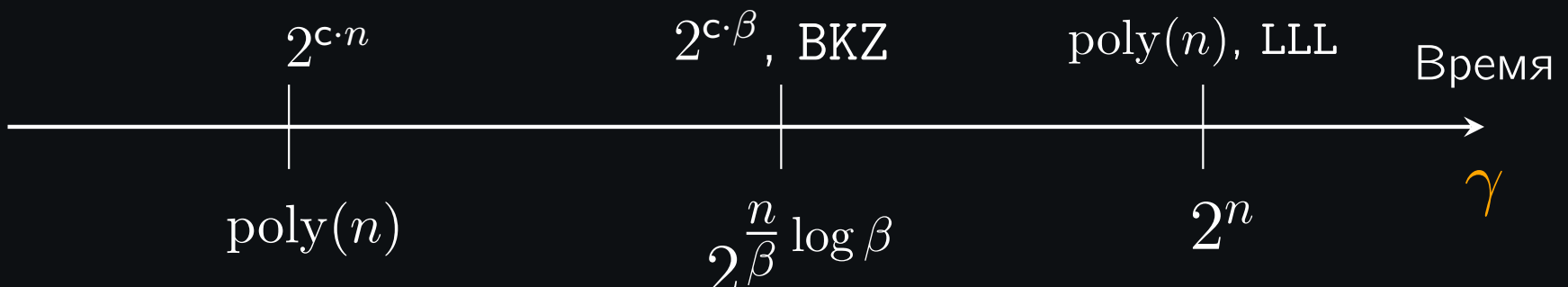
$$2^{\frac{m}{\beta} \log \beta} = \frac{q^{1-n/m}}{|e_i|},$$

и выбором $m = \Omega(n)$, минимизирующем решение.

Сложность LWE

LWE $(A, As + e)$ – BDD в $\mathcal{L}_q(A)$ с параметром

$$\gamma = \frac{\lambda_1(\mathcal{L}_q(A))}{\mathrm{dist}(As + e, \mathcal{L}_q(A))} = \frac{q^{1-n/m}}{|e_i|}$$



В асимптотике для константы c:

$$T(\mathrm{LWE}) = \exp\left(c \cdot \frac{\lg q}{\lg^2(q/|e_i|)} \lg\left(\frac{n \lg q}{\lg^2(q/|e_i|)}\right) \cdot n\right)$$

Эта формула получена решением уравнения для β

$$2^{\frac{m}{\beta} \log \beta} = \frac{q^{1-n/m}}{|e_i|},$$

и выбором $m = \Omega(n)$, минимизирующем решение.

Улучшения константы c и конкретные значений $T(\mathrm{LWE})$ – открытые вопросы криптоанализа.

Для грамотного выбора параметров

1. вычисляется асимтотическая сложность задачи (для каждого известного алгоритма)
2. алгоритм реализовывается на практике
3. асимптотика (множители малых порядков) уточняется интерполяцией
4. пишется скрипт для вычисления конкретного уровня безопасности для заданных параметров

Реальные SVP задачи

- TU Darmstadt предлагает решить задачи 1.05-SVP, LWE, `https://www.latticechallenge.org/svp-challenge/`
- Наиболее эффективным на сегодняшний день является алгоритм "просеивания" (sieving). Он реализован в опен-сорс проекте G6K `https://github.com/fplll/g6k`
- Этот алгоритм работает за время

$$\texttt{Time}(\text{SVP}) = 2^{0.349n+o(n)} \qquad \texttt{Memory}(\text{SVP}) = 2^{0.2075n+o(n)}.$$

Решения 1.05-SVP с помощью алгоритма G6K[3]

SVP dim	γ	Wall time	Total CPU time	RAM
155	1.00803	$14d\ 16h$	$1056d$	246 GiB
153	1.02102	$11d\ 15h$	$911d$	139 GiB
151	1.04411	$11d\ 19h$	$457.5d$	160 GiB
149	0.98506	$60h\ 7m$	$4.66kh$	59 GiB
147	1.03863	$123h\ 29m$	$4.79kh$	67.0 GiB
145	1.04267	$39h\ 3m$	$1496h$	37.7 GiB

[3] https://github.com/fplll/g6k

Часть IV

От решеток к кодам. Криптография на кодах

Решётки vs. Коды

$\mathcal{L}$	$\mathcal{C}$
Решетка $\mathcal{L}$ – аддитивная группа в $\mathbb{Z}^n$	Код $\mathcal{C}$ – аддитивная группа в $\mathbb{F}_2^n$
Евклидова метрика (ℓ_2)	ℓ_1 - метрика
$\|\mathbf{v}\|_2$	$wt(\mathbf{v}) = \|\{i : \mathbf{v}[i] > 0\}\|$- вес Хэмминга
$\lambda_1(\mathcal{L})$ - кратчайший вектор	$d(\mathcal{C})$-мин. расстояние
граница Минковского для $\lambda_1(\mathcal{L})$	граница Гильберта-Варшамова
$A \in \mathbb{Z}_q^{m \times n}$ – базис $\mathcal{L}_q(A)$	$G \in \mathbb{F}_2^{m \times n}$– порождающая матрица
Нормальное распределение	Распределение Бернулли

Решётки vs. Коды

$\mathcal{L}$	$\mathcal{C}$
Решетка $\mathcal{L}$ – аддитивная группа в $\mathbb{Z}^n$	Код $\mathcal{C}$ – аддитивная группа в $\mathbb{F}_2^n$
Евклидова метрика (ℓ_2)	ℓ_1 - метрика
$\|\mathbf{v}\|_2$	$wt(\mathbf{v}) = \|\{i : \mathbf{v}[i] > 0\}\|$- вес Хэмминга
$\lambda_1(\mathcal{L})$ - кратчайший вектор	$d(\mathcal{C})$-мин. расстояние
граница Минковского для $\lambda_1(\mathcal{L})$	граница Гильберта-Варшамова
$A \in \mathbb{Z}_q^{m \times n}$ – базис $\mathcal{L}_q(A)$	$G \in \mathbb{F}_2^{m \times n}$– порождающая матрица
Нормальное распределение	Распределение Бернулли

Трудные задачи

Задача BDD в $\mathcal{L}_q(A)$: Декодирование в $\mathcal{C}$:

Для $\mathbf{x} \in \mathbb{Z}_q(\mathbb{F}_2^n)$ найти $\mathbf{v} \in \mathcal{L}(\mathcal{C})$: $||\mathbf{x} - \mathbf{v}||_{2(1)}$ – минимально.

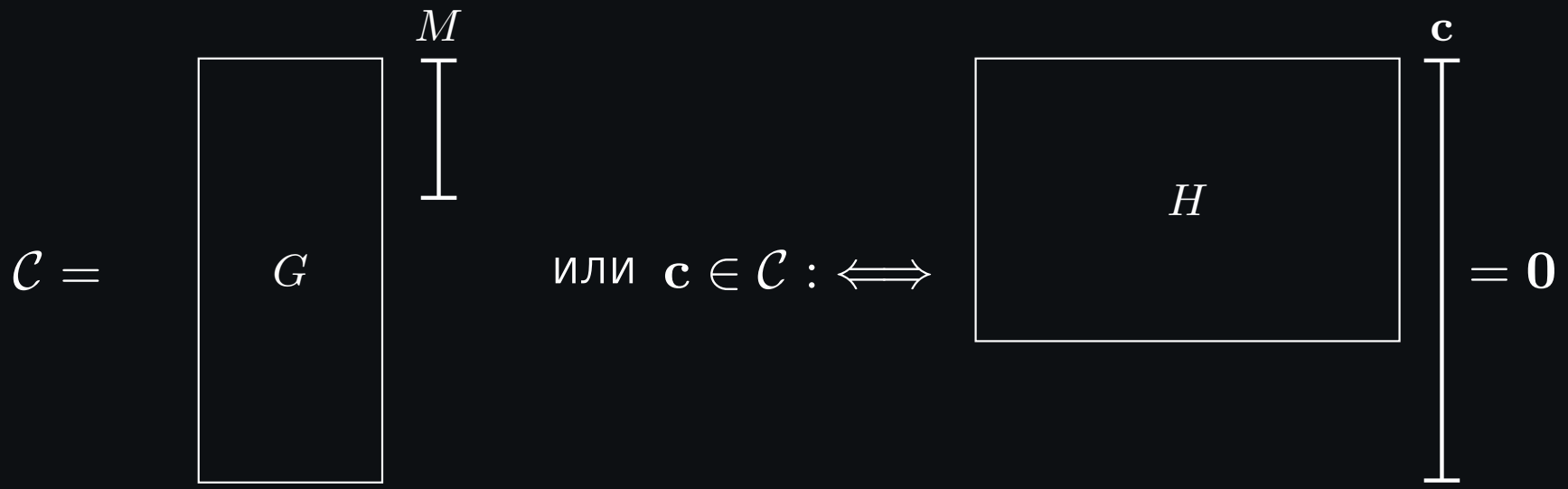
В крипто

Эффективные шифрование, подпись, много чего еще	Схема McEliece/Niederreiter подпись WAVE
Стойкие алгебраические структура	Большинство конструкций с алг. структурами взломаны

Задача декодирования

$G \in \mathbb{F}_2^{m \times n}$ – порождающая матрица $\mathcal{C}$

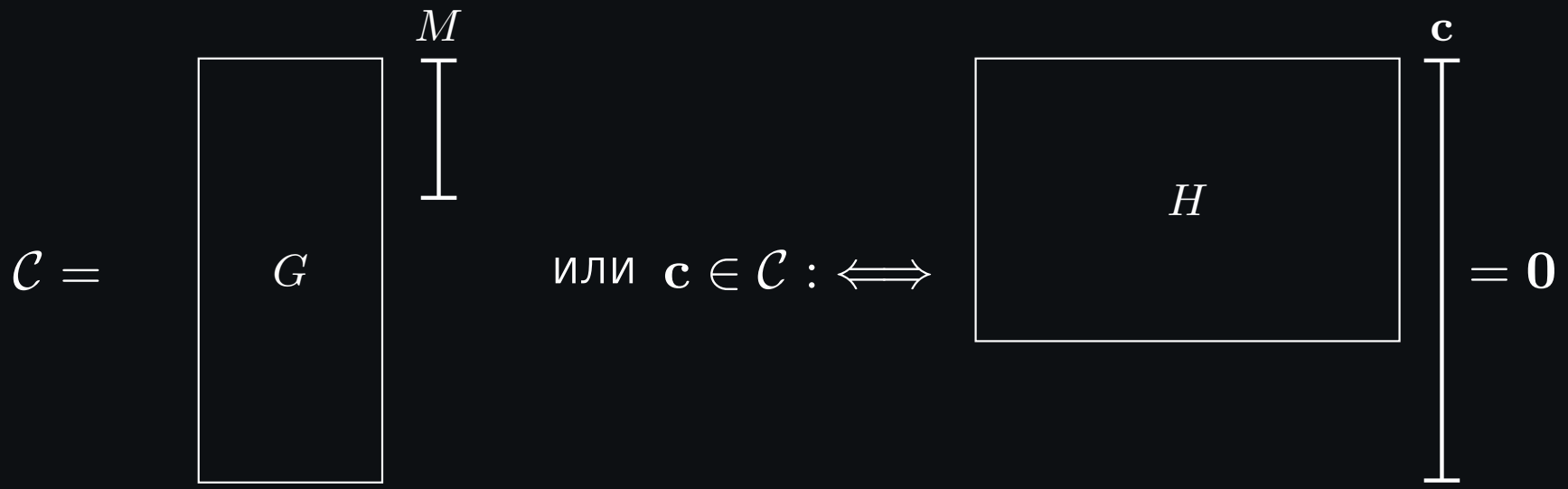
$H \in \mathbb{F}_2^{m-n \times m}$– проверочная матрица кода $\mathcal{C}$



Задача декодирования

$G \in \mathbb{F}_2^{m \times n}$ – порождающая матрица $\mathcal{C}$

$H \in \mathbb{F}_2^{m-n \times m}$– проверочная матрица кода $\mathcal{C}$

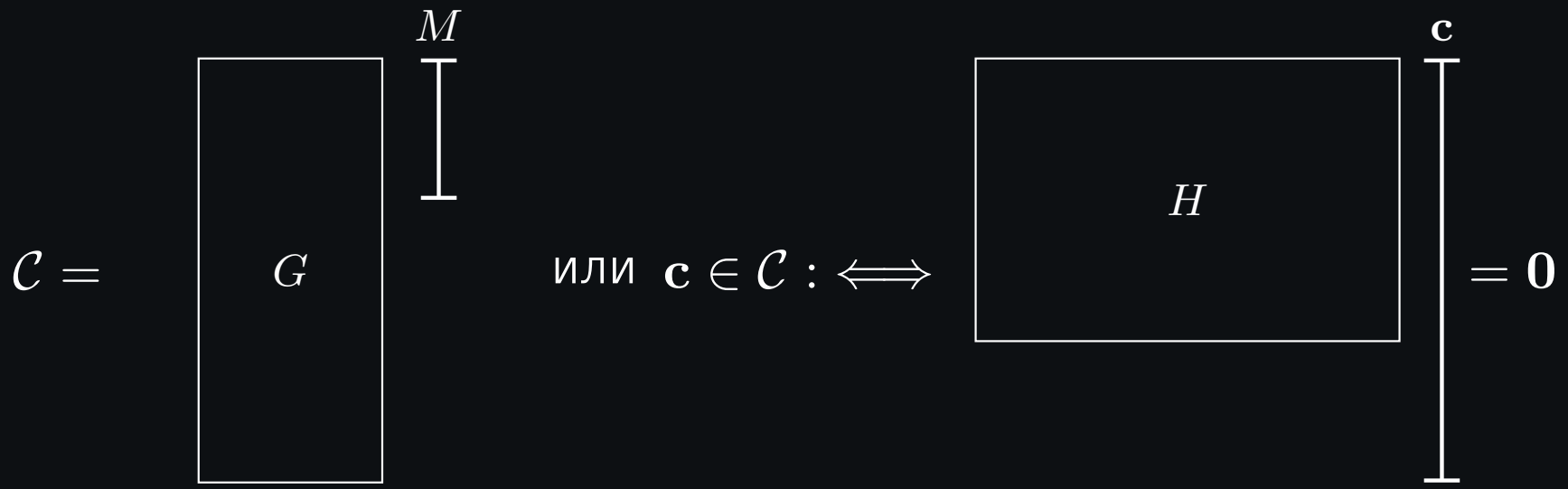


Задача декодирования: Для данных $H \in \mathbb{F}_2^{(m-n) \times m}$, $\mathbf{s} \in \mathbb{F}_2^{m-n}$ найти $\mathbf{e}$: $H\mathbf{e} = \mathbf{s}$ с $wt(\mathbf{e}) = w < d(\mathcal{C})$.

Параметры: $m = \Theta(n), d(\mathcal{C}) = \Theta(n)$.

Задача декодирования

$G \in \mathbb{F}_2^{m \times n}$ – порождающая матрица $\mathcal{C}$
$H \in \mathbb{F}_2^{m-n \times m}$– проверочная матрица кода $\mathcal{C}$



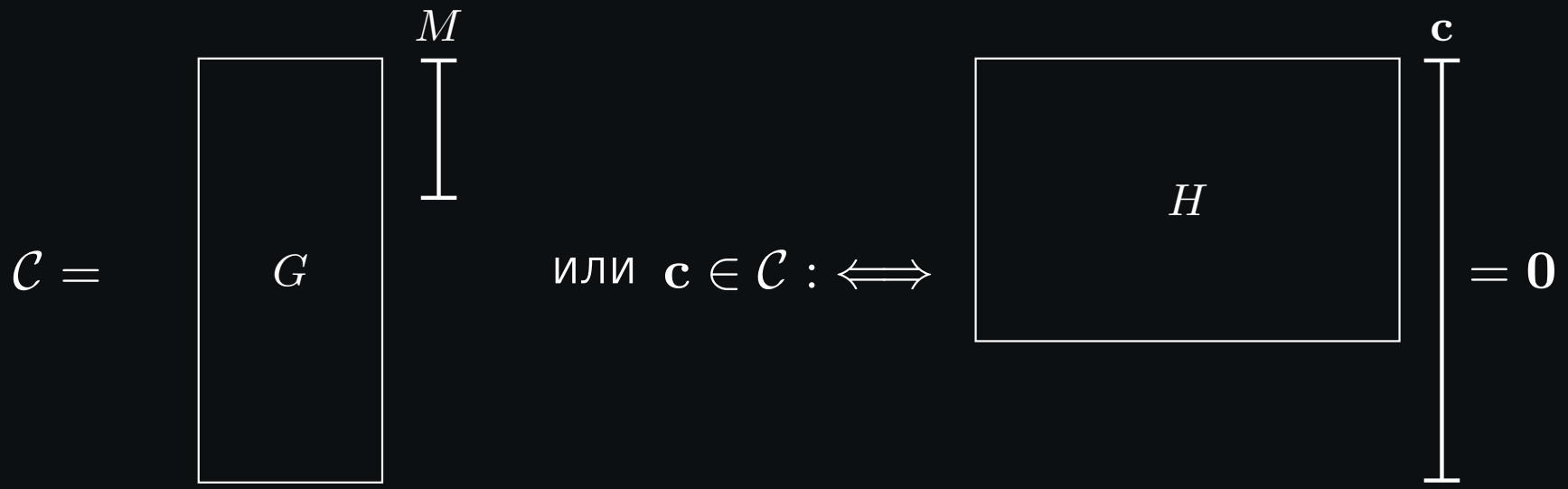
Задача декодирования: Для данных $H \in \mathbb{F}_2^{(m-n) \times m}$, $\mathbf{s} \in \mathbb{F}_2^{m-n}$
найти $\mathbf{e}$: $H\mathbf{e} = \mathbf{s}$ с $wt(\mathbf{e}) = w < d(\mathcal{C})$.

Параметры: $m = \Theta(n), d(\mathcal{C}) = \Theta(n)$.

- Для ‘структурированных’ матриц H (код Рида-Соломона, код Гоппы, ...), задача декодирования решается за время $\mathrm{poly}(n)$

Задача декодирования

$G \in \mathbb{F}_2^{m \times n}$ – порождающая матрица $\mathcal{C}$
$H \in \mathbb{F}_2^{m-n \times m}$– проверочная матрица кода $\mathcal{C}$



Задача декодирования: Для данных $H \in \mathbb{F}_2^{(m-n)\times m}$, $\mathbf{s} \in \mathbb{F}_2^{m-n}$
найти $\mathbf{e}$: $H\mathbf{e} = \mathbf{s}$ с $wt(\mathbf{e}) = w < d(\mathcal{C})$.

Параметры: $m = \Theta(n), d(\mathcal{C}) = \Theta(n)$.

- Для ‘структурированных’ матриц H (код Рида-Соломона, код Гоппы, ...), задача декодирования решается за время $\mathrm{poly}(n)$
- В крипто структура H “спрятана” случайным обратимым матрицами:

$$K = S \cdot H \cdot P,$$

$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$, $P \in \mathrm{GL}(m, \mathbb{F}_2)$– матрица перестановки

Сложность декодирования

Задача декодирования: Для данных $H \in \mathbb{F}_2^{(m-n)\times m}$, $\mathbf{s} \in \mathbb{F}_2^{m-n}$ найти $\mathbf{e}$: $H\mathbf{e} = \mathbf{s}$ с $wt(\mathbf{e}) = w < d(\mathcal{C})$.

- В худшем случае задача декодирования линейного кода – NP-сложная задача
- Для случайной $H \xleftarrow{\$} \mathbb{F}_2^{(m-n)\times m}$ и $wt(\mathbf{e}) = \Theta(n)$

$$T(\text{Decoding}) = 2^{\mathsf{c}\cdot n + o(n)}$$

- В крипто $wt(\mathbf{e}) = \Theta(n/\log(n))$:

$$T(\text{Decoding}) = 2^{\mathsf{c}\cdot \frac{n}{\log(n)} + o\left(\frac{n}{\log(n)}\right)}$$

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

- **KeyGen** :
  1. Выбираются случайные
  $$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$$
  $$P \in \mathrm{GL}(m, \mathbb{F}_2) - \text{ матрица перестановки}$$
  2. $K = S \cdot H \cdot P \in \mathbb{F}_2^{m-n\times m}$
  3. $\mathsf{pk} = K, \mathsf{sk} = (S, P)$

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

- **KeyGen** :
  1. Выбираются случайные
  $$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$$
  $$P \in \mathrm{GL}(m, \mathbb{F}_2) - \text{ матрица перестановки}$$
  2. $K = S \cdot H \cdot P \in \mathbb{F}_2^{m-n\times m}$
  3. $\mathsf{pk} = K, \mathsf{sk} = (S, P)$
- **Enc**$(\mathbf{pk}, \mathtt{mes} \in \{0,1\}^n, wt(\mathtt{mes}) = w)$

$$c = K \cdot \mathtt{mes} \in \mathbb{F}_2^{m-n}$$

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

- **KeyGen** :
  1. Выбираются случайные
  $$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$$
  $$P \in \mathrm{GL}(m, \mathbb{F}_2) - \text{ матрица перестановки}$$
  2. $K = S \cdot H \cdot P \in \mathbb{F}_2^{m-n\times m}$
  3. $\mathsf{pk} = K, \mathsf{sk} = (S, P)$
- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\}^n, wt(\mathtt{mes}) = w)$
$$c = K \cdot \mathtt{mes} \in \mathbb{F}_2^{m-n}$$
- **Dec**(sk, c):
  1. $\mathtt{mes}' = S^{-1}c = S^{-1}K \cdot \mathtt{mes} = S^{-1}SHP \cdot \mathtt{mes} = H \cdot (P \cdot \mathtt{mes})$. Декодирование толерантно к перестановка бит в $\mathtt{mes}$.

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

- **KeyGen** :
  1. Выбираются случайные
  $$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$$
  $$P \in \mathrm{GL}(m, \mathbb{F}_2) - \text{ матрица перестановки}$$
  2. $K = S \cdot H \cdot P \in \mathbb{F}_2^{m-n\times m}$
  3. $\mathsf{pk} = K, \mathsf{sk} = (S, P)$
- **Enc**$(\mathsf{pk}, \mathtt{mes} \in \{0,1\}^n, wt(\mathtt{mes}) = w)$
$$c = K \cdot \mathtt{mes} \in \mathbb{F}_2^{m-n}$$
- **Dec**(sk, c):
  1. $\mathtt{mes}' = S^{-1}c = S^{-1}K \cdot \mathtt{mes} = S^{-1}SHP \cdot \mathtt{mes} = H \cdot (P \cdot \mathtt{mes})$. Декодирование толерантно к перестановка бит в $\mathtt{mes}$.
  2. $P^{-1}(P \cdot \mathtt{mes}) = \mathtt{mes}$.

Схема McEliece/Niederreiter

Публичный параметр: $H \in \mathbb{F}_2^{(m-n)\times m}$ – проверочная матрица эффективного кода (код Гоппы).

- **KeyGen** :
  1. Выбираются случайные
  $$S \in \mathrm{GL}(m-n, \mathbb{F}_2)$$
  $$P \in \mathrm{GL}(m, \mathbb{F}_2) - \text{ матрица перестановки}$$
  2. $K = S \cdot H \cdot P \in \mathbb{F}_2^{m-n\times m}$
  3. $\mathsf{pk} = K, \mathsf{sk} = (S, P)$
- **Enc**$(\mathbf{pk}, \mathtt{mes} \in \{0,1\}^n, wt(\mathtt{mes}) = w)$

$$c = K \cdot \mathtt{mes} \in \mathbb{F}_2^{m-n}$$

Криптосистема безопасна под предположениями:

1. неотличимости K от случайной матрицы из $\mathbb{F}_2^{m-n\times m}$
2. сложности декодирования случайного кода

Конкретные параметры

В кандидате NIST McEliece[4] предложены параметры

$$m = 3488, \quad n = 2720, \quad wt(e) = 64.$$

Размеры параметров в байтах:

	$\mathcal{L}$ NewHope	$\mathcal{C}$ McEliece
$\|\text{pk}\|$	928	261120
$\|\text{sk}\|$	826	6452
$\|ct\|$	1088	128

- Преимущество McEliece: “Nothing has changed in more than 40 years in the asymptotics of OW-Passive security for McEliece.” [5]
- Недостаток: большие ключи, менее эффективные операции

[4] https://classic.mceliece.org/nist/mceliece-20190331.pdf
[5] https://classic.mceliece.org/talks/20190824.pdf

Открытые вопросы

1. Улучшенные алгоритмы SVP для “идеальных” решёток
2. Алгоритмы SVP для других норм, например, ℓ_∞.
3. Практическая реализация алгоритмов декодирования

Открытые вопросы

1. Улучшенные алгоритмы SVP для “идеальных” решёток
2. Алгоритмы SVP для других норм, например, ℓ_∞.
3. Практическая реализация алгоритмов декодирования

Больше о решетках и кодах:
`https://crypto-kantiana.com/`

?