# ИСПОЛЬЗОВАНИЕ ОРТОГОНАЛИЗАЦИИ ГРАМА-ШМИДТА В АЛГОРИТМЕ ПРИВЕДЕНИЯ БАЗИСА РЕШЕТКИ ДЛЯ ПРОТОКОЛОВ БЕЗОПАСНОСТИ

***Пискова А.В.[1], Менщиков А.А.[2], Коробейников А.Г.[3]***

*Информационные технологии в современном мире имеют огромное значение и широко применяются в различных областях. Информация, обрабатывающаяся в компьютерных сетях, должна быть хорошо защищена от широкого множества угроз. Это требует непрерывного совершенствования механизмов защиты информации. Согласно алгоритму Шора, с наступлением эры квантовых компьютеров, их огромная вычислительная мощность может вызвать сбой и компрометацию многих, используемых сегодня криптографических схем. Возникает актуальность поиска методов, устойчивых к квантовому криптоанализу. Одним из альтернативных подходов для решения данной проблемы является построение схем на основе сложности определенных свойств решеток, которые, как предполагается, будут неразрешимыми для квантовых компьютеров. Благодаря выдающимся научным достижениями в последние годы схемы на основе теории решеток уже стали использоваться на практике и рассматриваются как очень жизнеспособная альтернатива теоретико–числовой криптографии. В данной статье изучаются практические реализации алгоритмов решеток, которые используются при построении пост–квантовых протоколов безопасности. Проведен подробный анализ процесса ортогонализации Грама–Шмидта, являющегося одним из базовых и наиболее важных составляющих алгоритмов на решетках, примеры расчетов и вычислений, а также приводится его программная реализация. Дополнительно рассматривается алгоритм Ленстра–Ленстра–Ловаса приведения базиса решетки и его программная реализация, изучается его эффективность, точностные характеристики данного метода при решении задачи нахождения базиса, режимы и скорость работы в зависимости от размерности решетки, а также ключевые сферы и его практическое применение в современных протоколах безопасности.*

***Ключевые слова**: теория решеток, постквантовая криптография, целочисленные решетки, приведение базиса решетки, алгоритм Ленстра–Ленстра–Ловаса, LLL приведение решетки, квантовый компьютер, QR–разложение, ортогонализация, алгоритм Грама–Шмидта, схема NTRUEncrypt, шифрование.*

**Введение**

Почти все популярные на сегодняшний день криптографические схемы основываются на сложности классических трудных задач (факторизации, дискретного логарифмирования) [1], стойкость которых может быть серьезно ослаблена в случае достижений в классической криптоаналитике или прогресса в развитии квантовых компьютеров. Также многие криптографические алгоритмы уязвимы к атаке «человек посередине». Алгоритмы теории решеток с использованием базисов гораздо более устойчивы к атакам с помощью квантовых компьютеров, вследствие чего являются достойной заменой современным алгоритмам асимметричной криптографии.

Экспоненциальную точность $\gamma = 2^{(n-1)/2}$ в привидении базиса решетки к $(\delta - LLL)$-редуцированному базису дает алгоритм Ленстра-Ленстра-Ловаса (LLL), являющийся полиномиальным по временной сложности [2, с.120]. Этот алгоритм был впервые представлен в работе [3] в 1982 году учеными А. Ленстрой, Х. Ленстрой и Л. Ловасом. Далее рассмотрим его реализацию и зависимость времени его выполнения от размерности $\mathrm{N}$ решетки $\mathrm{L}$.

**Приведение базиса решетки**

Введем некоторые определения.

**Определение 1.** Решетка – это совокупность точек в $\mathrm{n}$-мерном пространстве с периодической структурой [4]. Более точно решетку $\mathrm{L}$ можно определить, как абелеву подгруппу, заданную в пространстве $\mathrm{R}^{\mathrm{m}}$.

Пусть базис решетки $B = \{b_1, \dots b_n\}$ задан линейно независимыми векторами, тогда под решеткой будем понимать множество целочисленных линейных комбинаций этих векторов

1 Пискова Антонина Владиславовна, Университет ИТМО, Санкт-Петербург, piter-ton@mail.ru.

2 Менщиков Александр Алексеевич, Университет ИТМО, Санкт-Петербург, menshikov@corp.ifmo.ru.

3 Коробейников Анатолий Григорьевич, доктор технических наук, профессор, Университет ИТМО, Санкт-Петербург, korobeynikov_a_g@mail.ru.

$$L(b_{1,\dots\dots,}b_n) = \{\sum_{i=1}^{n} a_i b_i : (a_1 \dots a_n) \in Z^n\} \quad (1)$$

Если ранг решетки (n) и размерность (m) равны, то решетка будет называться полноранговой.

**Определение 2**. Матрицей Грама называется матрица

$$\Gamma = \begin{Bmatrix} (b_1, b_1)\ (b_1, b_2) \dots \dots (b_1, b_n) \\ (b_2, b_1)\ (b_2, b_2) \dots \dots (b_1, b_n) \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ (b_n, b_1)\ (b_n, b_2) \dots \dots (b_n, b_n) \end{Bmatrix} \quad (2)$$

**Определение 3**. Определителем решетки называется число

$$\det L = \sqrt{\Gamma(b_1 \dots \dots b_n)}\,, \quad (3)$$

равное объему фундаментального параллелепипеда, натянутого на векторы $(b_1 \dots \dots b_n)$ [2] (см. рис. 1).

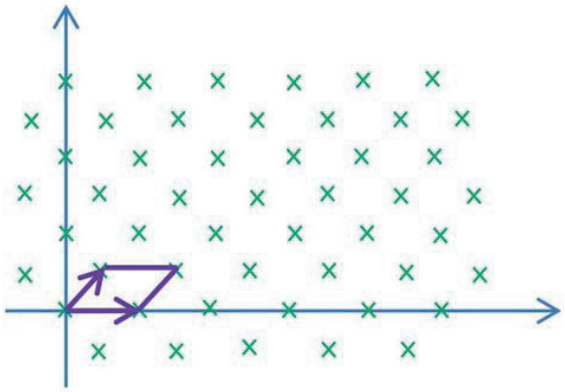

***Рис. 1**. Фундаментальный параллелепипед, образованный базисом*

Заметим, что если $m = n$, то $\det L = \det(b_1, \dots, b_n)$, где $(b_1, \dots, b_n)$ — матрица, составленная из координат векторов $b_1, \dots, b_n$ в некотором ортонормированном базисе. Базис решетки не единственен, но легко видеть, что матрица перехода от одного базиса решетки к произвольному другому унимодулярна, т. е. ее определитель равен ±1, поэтому $\det\Lambda$ не зависит от выбора базиса [5].

**Ортогонализация Грама-Шмидта**

Процесс ортогонализации является ключевой частью процесса приведения базиса решетки. В алгоритме LLL QR-разложение матрицы происходит с помощью рекуррентного алгоритма Грама-Шмидта. Напомним основные этапы этого процесса (см. рис.2).

Пусть $b_1, \dots, b_n$ линейно независимые векторы в $R^m$. Если векторы $b_1, \dots, b_n$ определяются соотношениями

$$\langle b_i^*, b_j \rangle = \langle b_i^*, b_i^* \rangle \quad (4)$$

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*\,, \qquad 1 \le i \le m \quad (5)$$

$$\text{где } \mu_{ij} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}, \qquad 1 \le j \le i \le n \quad (6)$$

тогда $\langle b_i^*, b_i \rangle = 0$ для всех $j < i$, то есть векторы $b_1^* \dots . b_n^*$ попарно ортогональны.

Далее приведен числовой пример работы процесса ортогонализации Грама-Шмидта:

1. Дана решетка с базисами $b_1$ = (5, 8, 11) и $b_2$ = (13, 9, 6), $b_3$ = (3, 14, 1).

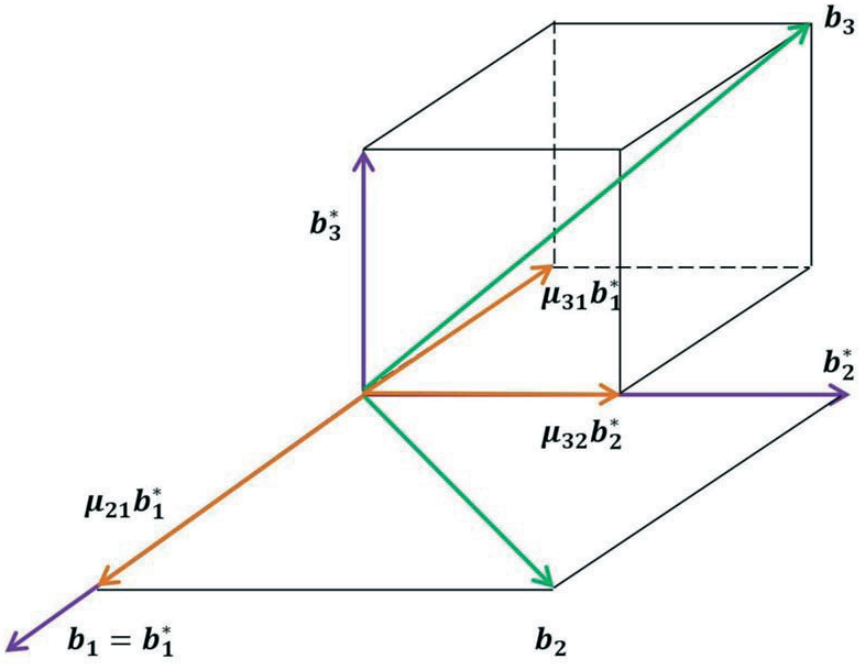


***Рис. 2.** Процесс ортогонализации Грама–Шмидта*

$$A = \begin{pmatrix} 5 & 13 & 3 \\ 8 & 9 & 14 \\ 11 & 6 & 1 \end{pmatrix}$$

2. Вычислим длину векторов $b_1$, $b_2$ и $b_3$ по формуле $||b_n|| = \sqrt{x^2 + y^2 + z^2}$ (7)

$$||b_1|| = \sqrt{5^2 + 8^2 + 11^2} \approx 14{,}49,$$

$$||b_2|| = \sqrt{13^2 + 9^2 + 6^2} \approx 16{,}91,$$

$$||b_3|| = \sqrt{3^2 + 14^2 + 1^2} \approx 14{,}35.$$

Примем $b_1^* = b_1$.

Вычислим коэффициенты $\mu_{ij}$ по формуле (6):

$$\mu_{21} = \frac{\langle b_2, b_1^* \rangle}{\langle b_1^*, b_1^* \rangle} = \frac{\langle 13*5 + 9*8 + 6*11 \rangle}{\langle 5*5 + 8*8 + 11*11 \rangle} = \frac{\langle \mathbf{203} \rangle}{\langle \mathbf{210} \rangle};$$

$$\mu_{31} = \frac{\langle b_3, b_1^* \rangle}{\langle b_1^*, b_1^* \rangle} = \frac{\langle 3*5 + 14*8 + 1*11 \rangle}{\langle 5*5 + 8*8 + 11*11 \rangle} = \frac{\langle \mathbf{138} \rangle}{\langle \mathbf{210} \rangle};$$

$$\mu_{32} = \frac{\langle b_3, b_2^* \rangle}{\langle b_2^*, b_2^* \rangle} = \frac{\langle 3*13 + 14*9 + 1*6 \rangle}{\langle 13*13 + 9*9 + 6*6 \rangle} = \frac{\langle \mathbf{171} \rangle}{\langle \mathbf{286} \rangle}.$$

Вычислим базисы $b_2^*$ и $b_3^*$ по формуле (5):

$$b_2^* = b_2 - \mu_{21} b_1^* = \left(13 - \frac{203}{210}*5; 9 - \frac{203}{210}*8; 6 - \frac{203}{210}*11\right) =$$

$$\left(\frac{\mathbf{343}}{\mathbf{42}}; \frac{\mathbf{133}}{\mathbf{105}}; -\frac{\mathbf{973}}{\mathbf{210}}\right) \approx (\mathbf{8.1667};\ \mathbf{1.2667};\ \mathbf{-4.6334});$$

$$b_3^* = b_3 - \mu_{31} b_1^* - \mu_{32} b_2^* = \left(3 - \frac{138}{210}*5 - \frac{171}{286}*\frac{343}{42}; 14 -\right.$$

$$\left. -\frac{138}{210}*8 - \frac{171}{286}*\frac{133}{105}; 1 - \frac{138}{210}*11 - \frac{171}{286}*-\frac{973}{210}\right) =$$

$$= \left(\frac{\mathbf{434595}}{\mathbf{84084}}; \frac{\mathbf{47961}}{\mathbf{6006}}; -\frac{\mathbf{41541}}{\mathbf{12012}}\right) \approx$$

$$\approx (\mathbf{-5.1685};\ \mathbf{7.9855};\ \mathbf{-3.4583}).$$

Вычислим длину векторов $b_1^*$, $b_2^*$ и $b_3^*$ по формуле (7):

$$||b_1^*|| = \sqrt{5^2 + 8^2 + 11^2} \approx 14{,}4914,$$

$$||b_2^*|| = \sqrt{\left(\frac{343}{42}\right)^2 + \left(\frac{133}{105}\right)^2 + \left(-\frac{973}{210}\right)^2} \approx 9{,}4745,$$

$$||b_3^*|| = \sqrt{\left(\frac{434595}{84084}\right)^2 + \left(\frac{47961}{6006}\right)^2 + \left(-\frac{41541}{12012}\right)^2} \approx 10{,}1214.$$

Можно видеть, что векторы $b_2^*$ и $b_3^*$ оказались короче исходных векторов. Ниже приведен исходный код на языке Python для выполнения расчетов процесса ортогонализации Грама-Шмидта. Данный алгоритм позволяет ортогонализовать вектора матрицы размера n*n в зависимости от исходного параметра N и количества координат базисных векторов b.

```
import math
from itertools import combinations
N = 3
b = [ [5, 8, 11], [13, 9, 6], [3, 14, 1] ]
assert len(b) == N
def vector_length(v):
    """
    Return vector length
    """
    return math.sqrt(sum([v[i]**2 for i in range(N)]))
def vector_string(v):
    """
    Print vector components
    """
    return ', '.join(map(lambda x: "%0.4f" % x, v))
if __name__ == "__main__":
    for i, v in enumerate(b):
          print "Vector #%d length: %0.4f" % (i + 1, vector_length(v))
    m = {}
    for i in range(N):
          for j in range(i):
                vector_i = b[i]
                vector_j = b[j]

                v = sum([c[0]*c[1] for c in zip(vector_i, vector_j)])
                u = sum([c[0]*c[1] for c in zip(vector_j, vector_j)])
                m[(i, j)] = v*1.0/u
                print "Coefficient (%d,%d) = %d/%d = %0.4f" % (i+1, j+1, v, u, v*1.0/u)
```

```
b_new = [b[0]]
print "b_1* = b_1 = (%s)" % (vector_string(b[0]))
for i in range(1, N):
      print "b_%d* = b_%d" % (i+1, i+1),
      s = ""
      for j in range(i):
            s += "- m_%d%d * b_%d* " % (i+1, j+1, j+1)

      vector_new = [b[i][a] for a in range(N)]
      for a in range(N):
            for j in range(i):
                  vector_new[a] -= m[(i, j)] * b_new[j][a]
      b_new.append(vector_new)

      print "%s = (%s)" % (s, vector_string(vector_new))

for i, v in enumerate(b_new):
      print "New vector #%d length: %0.4f" % (i + 1, vector_length(v))
```

**Результат выполнения программы**

```
Vector #1 length: 14.4914
Vector #2 length: 16.9115
Vector #3 length: 14.3527
Coefficient (2,1) = 203/210 = 0.9667
Coefficient (3,1) = 138/210 = 0.6571
Coefficient (3,2) = 171/286 = 0.5979
b_1* = b_1 = (5.0000, 8.0000, 11.0000)
b_2* = b_2 - m_21 * b_1* = (8.1667, 1.2667, -4.6333)
b_3* = b_3 - m_31 * b_1* - m_32 * b_2* = (-5.1686, 7.9855, -3.4583)
New vector #1 length: 14.4914
New vector #2 length: 9.4745
New vector #3 length: 10.1214
```

Как можно видеть, результаты обоих вычислений совпадают. Время, затраченное на выполнение данного расчета - 0.0000350 сек. Этот показатель велик по сравнению с теми, что будут освещены в следующем разделе, поскольку реализация алгоритма Грама-Шмидта выполнена непосредственно на языке Python, в то время как алгоритм LLL реализован с использованием предкомпилированных модулей из библиотеки NumPy, которые выполняются существенно быстрее.

**Алгоритм Ленстра-Ленстра-Ловаса**

Алгоритм Ленстра-Ленстра-Ловаса приведения базиса решетки является довольно популярным на сегодняшний день и в открытом доступе можно найти некоторые его модификации. В данной работе использована реализация, основанная на открытых математических библиотеках из проектов [6] и [7].

Для оценки зависимости времени выполнения алгоритма от размерности решетки N (См. Рис. 3) были взяты матрицы размерностью от 1 до 100 с шагом 1 при стократной точности вычислений. В расчет были взяты матрицы, состоящие из случайных рациональных значений в диапазонах от -100 до 0, от -100 до 100, от 0 до 100, а также матрицы рациональных чисел в диапазоне от -0.0000001 до 0.0000001 округлением до 7 знаков после запятой. Все вычисления проводились на персональном компьютере с процессором Intel Core i5-3317U.

Из графика видно, что время, затраченное на выполнение алгоритма полиномиально зависит от размерности решетки. Следует также отметить, что с ростом размерности решетки, рост необходимой точности представления чисел с плавающей запятой невысок для ансамблей случайных решеток.

Таким образом, современные реализации алгоритма являются достаточно эффективными по времени, использованию памяти и процессорных ресурсов. Скорость выполнения вычислений позволяет считать матрицы больших размерностей в широким наборе рациональных значений.

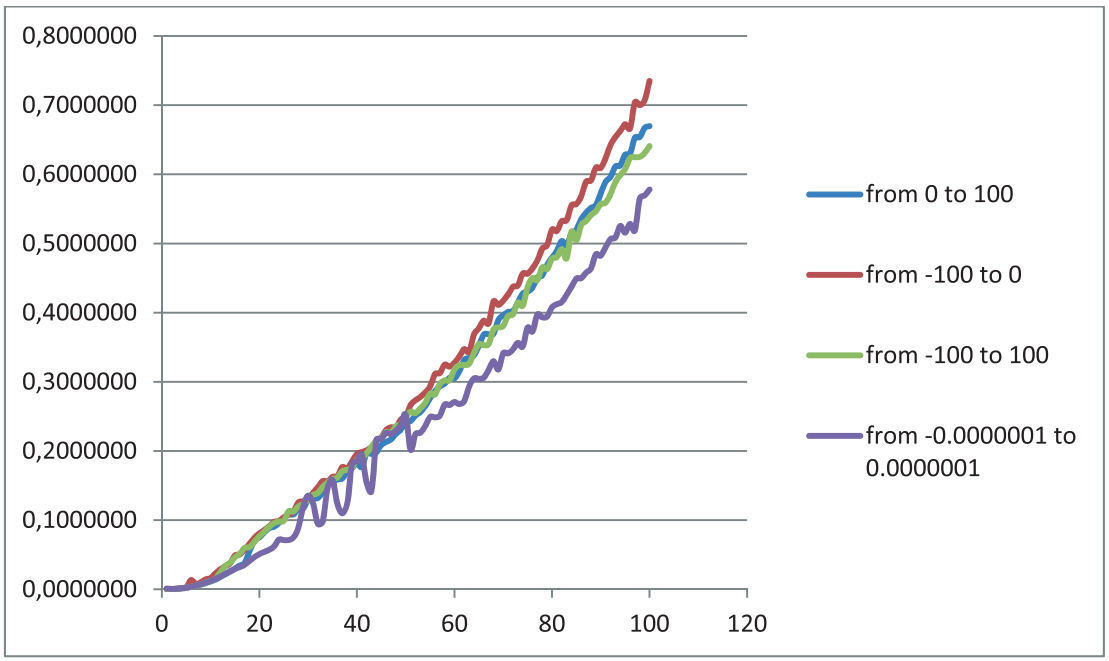


***Рис. 3**. Зависимость времени QR-разложения от размера базиса решетки*

**Применение алгоритма LLL**

Первоначально авторы работы [3] А. Ленстра, Х. Ленстра и Л. Ловас использовали LLL-алгоритм для факторизации многочленов с целыми коэффициентами. Также алгоритм был взят в основу для построения схемы NTRUEncrypt, которая, в свою очередь, была создана как альтернатива системам, основанным на задачах факторизации, дискретного логарифмирования и дискретного логарифмирования на эллиптических кривых [8]. Кроме того, различные реализации LLL-алгоритма были использованы в технологии MIMO, в кросс-платформенных системах компьютерной алгебры GAP, Magma, Maple, Sage и других.

**Выводы**

Таким образом, в работе был выполнен анализ алгоритма Ленстра-Ленстра-Ловаса приведения базиса решетки, произведен расчет процесса ортогонализации Грама-Шмидта и выполнена соответствующая программная реализация. Приведены точностные характеристики метода при решении задачи нахождения базиса в целочисленной решетке. Рассмотрены сферы применения данного алгоритма. Результаты, полученные в ходе работы, могут найти важное практическое применение в сферах информационного обмена в постквантовом мире.

**Литература:**

1. Пискова А.В. Разработка алгоритма электронной цифровой подписи, основанного на задачах факторизации и дискретного логарифмирования на эллиптических кривых // В сборнике: Сборник трудов IV Всероссийского конгресса молодых ученых Университет ИТМО, 2015. С. 319-322.
2. Усатюк В.С. Реализация параллельных алгоритмов ортогонализации в задаче поиска кратчайшего базиса целочисленных решеток // Прикладная дискретная математика. Приложение. 2012. № 5. С. 120-122.
3. Lenstra A. K., Lenstra H.W., Lov´asz L. Factoring Polynomials with Rational Coefficients // Math. Annalen, 1982, vol. 261, pp. 515–534.
4. Howe, J., Pöppelmann, T., O'Neill, M., O'Sullivan, E., Güneysu, T.: Practical Lattice - based Digital Signature Schemes. ACM Transactions on Embedded Computing Systems (ACM T EMBED COMPUT S). 14(3), 1-24 (2015)
5. Кузьмин О.В., Усатюк В.С. Параллельные алгоритмы вычисления локальных минимумов целочисленных решеток // Программные продукты и системы. 2015. № 1 (109). С. 55-62.
6. NumPy, URL: http://www.numpy.org (дата обращения: 25.01.16).
7. Kwant Project, URL: http://kwant-project.org (дата обращения: 25.01.16).
8. Пискова А.В. Разработка комбинированной схемы аутентификации информации, основанной на задачах факторизации и дискретного логарифмирования на эллиптических кривых. Аннотированный сборник научно-исследовательских выпускных квалификационных работ специалистов Университета ИТМО / Главный редактор Проректор по НР д.т.н., профессор В.О. Никифоров. – СПб: Университет ИТМО, 2015. – С.39–42.

**Рецензент**: Исмагилов Валерий Сарварович, кандидат физико-математических наук, ivs@izmiran.spb.ru

 УДК 004.056.5 + 512.545

# USE OF GRAM-SHMIDT ORTHOGONALIZATION IN THE LATTICE BASIS REDUCTION ALGORITHM FOR SECURITY PROTOCOLS

*Piskova A.V.[4], Menshchikov A.A.[5], Korobeynikov A.G.[6]*

*Information technologies have significant value in the modern world and are widely applied in various areas. Information which is processed in computer networks has to be well protected against wide varity of threats. This requires a constant improvement of the data protection's system. According to Shor's algorithm, with approach of a quantum computer era, the huge computing power can cause failure and compromise of many cryptographic schemes used today. There is a need to find methods that are resistant to quantum crypt–analysis. One of the alternative approaches for this problem solution is the creation of schemes based on the certain lattice complexity characteristics. These characteristics are supported to be unsolved for quantum computers. Due to the recent considerable scientific achievements, lattice schemes have already begun to be used in practice and seem to be a very viable alternative to the number–theoretic cryptography. In this article we provide practical implementations of the lattice algorithms which are used during the construction of a post–quantum security protocol. We carry out the detailed analysis of Gram–Schmidt orthogonalization process, which is one of the basic and most important part of the lattice algorithms, equations examples and its programming implementation. Additionally we consider the LLL lattice reduction algorithm analysis and its programming implementation, its efficiency, performance characteristics during the basis resolution pro–cess, modes and performance depending on a lattice dimension as well as the main scopes of the algorithm and the practical application in security protocols.*

***Keywords**: theory of lattices, post–quantum cryptography, integer lattices, lattice basis reduction, Lenstra–Lenstra–Lovasz algorith, LLL reduction, quantum computer, QR decomposition, orthogonalization, Gram–Schmidt algorithm, NTRUEncrypt, encryption*

**References**:

1. Piskova A.V. Razrabotka algoritma elektronnoy tsifrovoy podpisi, osnovannogo na zadachakh faktorizatsii i diskretnogo logarifmirovaniya na ellipticheskikh krivykh, V sbornike: Sbornik trudov IV Vserossiyskogo kongressa molodykh uchenykh Universitet ITMO, 2015, pp. 319-322.
2. Usatyuk V.S. Realizatsiya parallel'nykh algoritmov ortogonalizatsii v zadache poiska kratchayshego bazisa tselochislennykh reshetok, Prikladnaya diskretnaya matematika. Prilozhenie. 2012. No 5, pp. 120-122.
3. Lenstra A. K., Lenstra H.W., Lov´asz L. Factoring Polynomials with Rational Coefficients, Math. Annalen, 1982, vol. 261, pp. 515–534.
4. Howe, J., Pöppelmann, T., O'Neill, M., O'Sullivan, E., Güneysu, T.: Practical Lattice - based Digital Signature Schemes. ACM Transactions on Embedded Computing Systems (ACM T EMBED COMPUT S). 14(3), 1-24 (2015)
5. Kuz'min O.V., Usatyuk V.S. Parallel'nye algoritmy vychisleniya lokal'nykh minimumov tselochislennykh reshetok, Programmnye produkty i sistemy. 2015. No 1 (109), pp. 55-62.
6. NumPy, URL: http://www.numpy.org (data obrashcheniya: 25.01.16).
7. Kwant Project, URL: http://kwant-project.org (data obrashcheniya: 25.01.16).
8. Piskova A.V. Razrabotka kombinirovannoy skhemy autentifikatsii informatsii, osnovannoy na zadachakh faktorizatsii i diskretnogo logarifmirovaniya na ellipticheskikh krivykh, Annotirovannyy sbornik nauchno-issledovatel'skikh vypusknykh kvalifikatsionnykh rabot spetsialistov Universiteta ITMO / Glavnyy redaktor Prorektor po NR d.t.n., professor V.O. Nikiforov. – SPb: Universitet ITMO, 2015, – pp.39–42.

4 Antonina Piskova, St. Petersburg National Research University of Information Technologies, St. Petersburg, piter-ton@mail.ru;

5 Aleksandr Menshchikov, St. Petersburg National Research University of Information Technologies, St. Petersburg, menshikov@corp.ifmo.ru;

6 Anatoly Korobeynikov, Dr.Sc., Professor, St. Petersburg National Research University of Information Technologies, St. Petersburg, korobeynikov_a_g@mail.ru