УДК 621.391

ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМА ОБУЧЕНИЯ С ОШИБКАМИ АЛГЕБРАИЧЕСКИХ РЕШЕТЧАТЫХ КОДОВ

М.А. АЛИСЕЕНКО, С.Б. САЛОМАТИН

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 10 марта 2021

Аннотация. Рассмотрен алгоритм обучения с ошибками LWE для алгебраических решетчатых кодов. Приведена реализация алгоритма LWE на языке программирования Python. Показана временная сложность вычисления алгоритма LWE для различных длин открытого сообщения и наличия ошибок.

Ключевые слова: алгоритм обучения с ошибками, LWE, алгебраические решетки.

Введение

Одной из задач разработки алгоритмов защиты данных является их потенциальная способность противостоять различного вида атакам, в том числе на основе пост-квантовых и параллельных вычислений. Применение алгоритмов теории многомерных алгебраических решеток предоставляют возможность формирования пространственно-временного многообразия кодовых криптографических структур [1–4]. В данной работе проведена оценка временной сложности алгоритма обучения с ошибками LWE на основе теории решеток.

Моделирование алгоритма обучения с ошибками алгебраических решетчатых кодов

Алгоритм обучения с ошибками LWE [5]:

1. Алгоритм генерация ключа LWE. Вход: $LWE = n,\ m,\ l,\ q$ – целые числа.

1.1. Выбрать секретный ключ $S \in \mathbb{Z}_q^{n\times l}$ случайным образом.

1.2. Выбрать открытый ключ $A \in \mathbb{Z}_q^{m\times n}$ случайным образом.

1.3. Выбрать $E \in \mathbb{Z}_q^{m\times l}$ согласно χ.

1.4. Вычислить открытый ключ $P = AS + E(\mathrm{mod}\, q)$, где $P \in \mathbb{Z}_q^{m\times l}$.

Выход: закрытый ключ S и открытый ключ $(A;P)$.

2. Алгоритм шифрования. Вход: целые числа $n,\ m,\ l,\ t,\ r,\ q$, открытый ключ $(A;P)$, открытый текст $M \in \mathbb{Z}_t^{l\times 1}$.

2.1. Выбрать $a \in [-r,r]^{m\times 1}$ случайным образом.

2.2. Вычислить шифротекст $u = A^T a(\mathrm{mod}\, q) \in \mathbb{Z}_q^{n\times 1}$.

2.3. Вычислить шифротекст $c = P^T a + [Mq/t](\mathrm{mod}\, q) \in \mathbb{Z}_q^{l\times 1}$.

Выход: шифротекст (u,c).

3. Алгоритм расшифрования. Вход: целые $n,\ m,\ l,\ t,\ r,\ q$, секретный ключ S, шифротекст (u,c).

3.1. Вычислить $v = c - S^T u$ и $M = [tv/q]$.

Выход: открытый текст M.

Алгоритм реализован на Python 3.8.7 с использованием модуля numpy. Среднее время вычислений рассчитано из 20 измерений на каждую длину открытого текста с использованием встроенного модуля time. При вычислениях использовались следующие параметры:

$n = 3,\ m = 3,\ t = 10,\ r = 9,\ q = 23$, длина сообщения l, состоящего из случайных целых чисел от 0 до r, варьировалась от 2^1 до 2^{20}. Код функции алгоритма представлен ниже.

```
import numpy as np
import random
import time
def algorithm_lwe(n, m, l, t, r, q, e):
    t0 = time.time()
    lwe_message = np.array([random.randint(0, r) for index in range(l)])
    secret_key = np.random.randint(q, size=(n, l))
    public_key_a = np.random.randint(q, size=(m, n))
    errors = np.zeros((m, l))
    errors = add_errors(errors, e)
    public_key_p = np.mod(((np.dot(public_key_a, secret_key)) + errors), q)
    a_column = np.array([random.randint(-r, r) for index in range(m)])
    ciphertext_u = np.mod(np.dot(public_key_a.transpose(), a_column), q)
    c = np.mod(np.dot(public_key_p.transpose(), a_column) + np.dot(lwe_message, q) / t, q)
    v = np.mod((c - np.dot(secret_key.transpose(), ciphertext_u)), q)
    decoded_message = np.dot(t, v) / q
    message_verification = lwe_message - np.around(decoded_message)
    t1 = time.time()
    print(t1-t0)
```

Результаты вычислений представлены в таблице и на рис. 1 (график построен использованием модуля matplotlib). Сложность вычислений растет экспоненциально.

Длина открытого текста и среднее время вычислений

Длина открытого текста	Среднее время вычислений, с					
	0 ошибок	1 ошибка	2 ошибки	3 ошибки	4 ошибки	5 ошибок
2^1	менее 0,00077692	менее 0,00078103	менее 0,0007809	менее 0,00078601	менее 0,00078105	0,00078129
2^2	менее 0,00077692	менее 0,00078103	0,00078129	менее 0,00078601	менее 0,00078105	0,0007811
2^3	менее 0,00077692	менее 0,00078103	0,0007809	0,00078601	0,00078105	0,00078131
2^4	менее 0,00077692	менее 0,00078103	0,00078121	0,00078114	0,00078108	0,00156629
2^5	0,00077692	0,00078103	0,00078155	0,0	0,00078124	0,0
2^6	0,00078605	0,00078129	0,00156257	0,00078124	0,00078578	0,00077728
2^7	0,00156108	0,00078106	0,00078126	0,00077665	0,00077662	0,00078146
2^8	0,00156344	0,00156243	0,00312494	0,0015626	0,00078135	0,0023437
2^9	0,00233914	0,00156246	0,00390598	0,00234364	0,0023435	0,00234365
2^{10}	0,00547355	0,00625469	0,00859376	0,00547352	0,0046881	0,00781653
2^{11}	0,0124952	0,01093295	0,01406243	0,00858892	0,00859332	0,01249608
2^{12}	0,01718754	0,01796856	0,0367188	0,02500415	0,01797349	0,02812887
2^{13}	0,04140624	0,04531261	0,05703115	0,0546874	0,04062026	0,053066
2^{14}	0,08984395	0,0851567	0,10781269	0,11249609	0,08203155	0,1195312
2^{15}	0,17187498	0,17500001	0,22109404	0,21796908	0,18984798	0,1859421
2^{16}	0,34599365	0,3664056	0,40843289	0,45463223	0,3748706	0,3507766
2^{17}	0,66953600	0,7117192	0,87031239	0,8585941	0,75546863	0,7506607
2^{18}	1,37562145	1,5239129	1,66620113	1,64004219	1,5120651	1,5125001
2^{19}	2,79823248	3,0875399	3,4039516	2,93371499	3,05841386	3,1083119
2^{20}	5,66675596	6,3474723	6,96240778	6,06066538	6,66182725	6,2066043

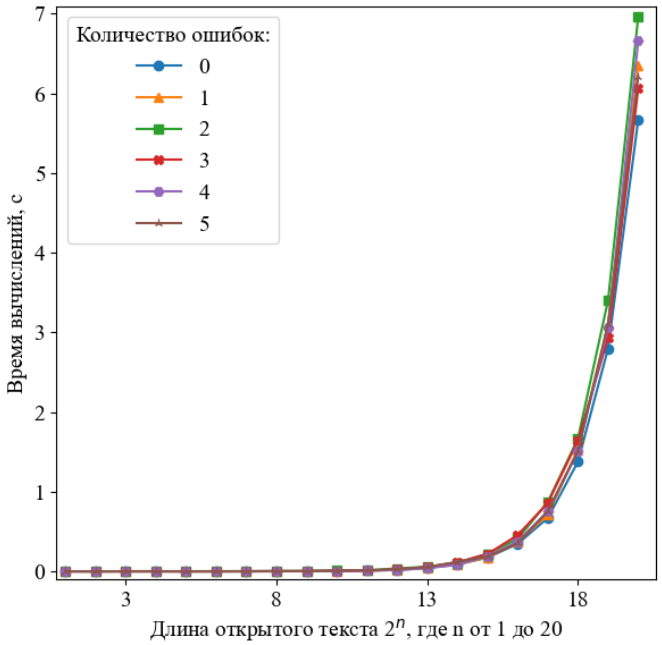

Рис. 1. График зависимости среднего времени вычислений в секундах
от количества символов открытого текста и наличия ошибок

Заключение

В настоящее время важность задачи разработки алгоритмов защиты данных определена их потенциальной способностью противостояния к разного рода атакам, таким как постквантовые и параллельные вычисления. Актуальной проблемой становится поиск быстрых алгоритмов шифрования с минимальной вычислительной сложностью для систем связи в сенсорных сетях. Временная сложность алгоритма обучения с ошибками LWE алгебраических решетчатых кодов растет экспоненциально по мере увеличения длины открытого текста, что необходимо учитывать в аппаратном обеспечении устройств интернета вещей.

COMPLEXITY ESTIMATION OF THE LEARNING WITH ERRORS ALGORITHM FOR ALGEBRAIC LATTICE CODES

M.A. ALISEYENKA, S.B. SALOMATIN

Abstract. The LWE (Learning with errors) algorithm for algebraic lattice codes is considered. The implementation of the LWE algorithm in Python is given. The time complexity of calculating the LWE algorithm for different open message lengths and the presence of errors is shown.

Keywords: learning with errors, LWE, algebraic lattice.

Список литературы

1. Ferdinand N.S. // Low complexity lattice codes for communication networks. University of Oulu Graduate School, 2016. P. 178.
2. Olds C.D. // The Geometry of Numbers. Mathematical Association of USA, 2012. P. 192.
3. Johnson N.W., Weiss A.I. // Canadian Journal of Mathematics. 1999.
4. Stallings W. Cryptography and Network Security: Principles and Practic. Prentice-Hall, Upper Saddle River, New-Jersey, fifth edition, 2006.
5. Алисеенко М.А., Саломатин С.Б. // Телекоммуникации: сети и технологии, алгебраическое кодирование и безопасность данных: материалы междунар. науч.-техн. семинара (Республика Беларусь, Минск, ноябрь – декабрь 2020 г.). Минск: БГУИР, 2020. С 23–27.