1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. Институт компьютерных наук и кибербезопасности
4. Высшая школа кибербезопасности

**ЛАБОРАТОРНАЯ РАБОТА № 6**

1. «Вычисление кратной точки эллиптической кривой»
2. по дисциплине «Быстрые вычислительные алгоритмы»
3. Выполнил
4. студент гр. 5151001/00201 Устюгов А.А.

<*подпись*>

1. Преподаватель д.т.н. Шенец Н.Н.

<*подпись*>

1. Санкт-Петербург
2. 2024

# Цель работы

Изучение алгоритмов вычисления кратной точки эллиптической кривой

# Задание

Написать программу, реализующую бинарный алгоритм вычисления кратной точки эллиптической кривой, а также алгоритм из варианта задания. Провести эксперименты с замерами времени вычисления кратной точки эллиптической кривой реализованными алгоритмами.

# Ход работы

Программа написана на C++ с использованием библиотеки NTL. Вариант 6: удвоение в координатах Чудновского, сложение в смешанных чудновско-аффинных координатах.

Был написан класс эллиптической кривой в краткой форме Вейерштрасса, а также класс точки. Класс кривой принимает характеристику поля и коэффициенты A, B. Точку можно задать как аффинными координатами, так и сразу в координатах Чудновского. Также можно задать как бесконечно удаленную.

Интерфейс программы представлен на рисунке 1.

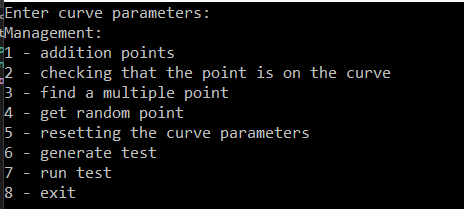


Рисунок 1 – интерфейс программы

Был реализован бинарный алгоритм вычисления кратной точки эллиптической кривой справа налево. Для этого число k представляется в двоичном виде, и, начиная с младших бит, проверяется значение очередного бита, если он равен 1, то складываем точки и удваиваем, иначе только складываем.

По варианту задания мне необходимо реализовать скользящий оконный NAF, однако был написан метод, вычисляющий . Поэтому, были реализованы следующие алгоритмы: NAF, window NAF, sliding window NAF. Идея алгоритма window NAF состоит в том, что вычисляется и вычисляются окна фиксированной длины *w*. Если окно ненулевое, то происходит удваивание и сложение с предвычисленной точкой, к которой можно обратиться с помощью значения окна. Иначе, только удваивание. Sliding window NAF использует обычное NAF представление, и сам выбирает размер окна, не превышающий заданный. NAF это sliding NAF с *w*=1.

Далее, на рисунке 2 представлен результат контрольного теста

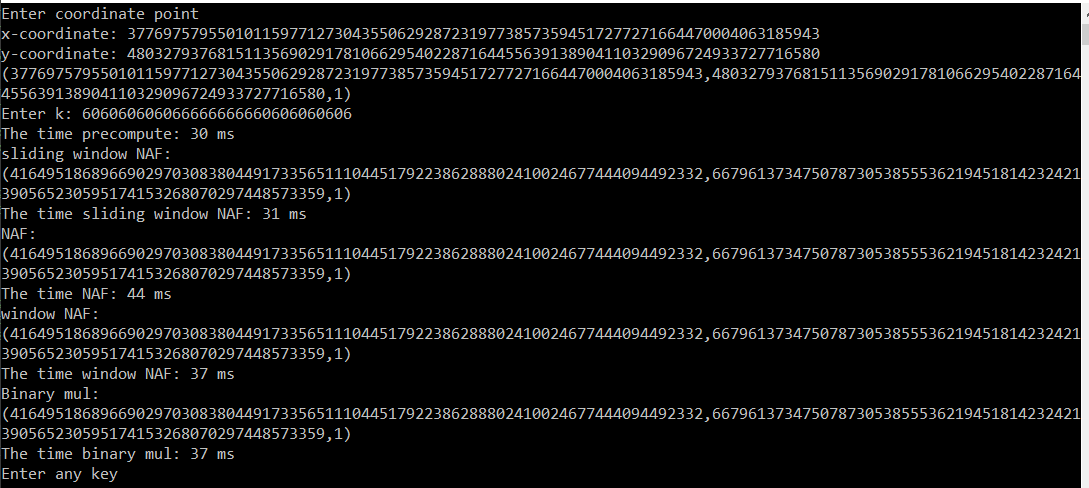


Рисунок 2 – тест из варианта

Далее приведены таблицы при различных k и w. В таблице приведены средние значения затраченного времени на вычисление кратной точки из 500 примеров. Время в миллисекундах.

Таблица 1 – время вычисления кратной точки при w=3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм  Длина k | Бинарный алгоритм | NAF | Window NAF | Sliding window NAF |
| 50 | 17 | 19 | 17 | 17 |
| 75 | 27 | 28 | 26 | 27 |
| 100 | 40 | 59 | 51 | 40 |
| 125 | 70 | 82 | 79 | 74 |
| 150 | 61 | 77 | 78 | 76 |
| 175 | 65 | 69 | 66 | 68 |
| 200 | 74 | 79 | 76 | 78 |
| 250 | 94 | 97 | 93 | 98 |

Таблица 2 – время вычисления кратной точки при w=4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм  Длина k | Бинарный алгоритм | NAF | Window NAF | Sliding window NAF |
| 50 | 17 | 18 | 18 | 16 |
| 75 | 27 | 28 | 25 | 26 |
| 100 | 34 | 38 | 33 | 33 |
| 125 | 46 | 49 | 43 | 44 |
| 150 | 56 | 58 | 52 | 54 |
| 175 | 66 | 68 | 62 | 62 |
| 200 | 77 | 78 | 71 | 73 |
| 250 | 94 | 97 | 87 | 90 |

Таблица 3 – время вычисления кратной точки при w=5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм  Длина k | Бинарный алгоритм | NAF | Window NAF | Sliding window NAF |
| 50 | 18 | 19 | 15 | 16 |
| 75 | 27 | 28 | 25 | 24 |
| 100 | 37 | 38 | 33 | 35 |
| 125 | 46 | 49 | 42 | 42 |
| 150 | 55 | 57 | 51 | 50 |
| 175 | 64 | 67 | 60 | 60 |
| 200 | 75 | 78 | 68 | 67 |
| 250 | 94 | 98 | 87 | 84 |

Таблица 4 – время вычисления кратной точки при w=6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм  Длина k | Бинарный алгоритм | NAF | Window NAF | Sliding window NAF |
| 50 | 18 | 19 | 16 | 15 |
| 75 | 27 | 28 | 24 | 23 |
| 100 | 37 | 38 | 33 | 31 |
| 125 | 49 | 49 | 41 | 40 |
| 150 | 56 | 59 | 52 | 48 |
| 175 | 66 | 69 | 59 | 56 |
| 200 | 76 | 77 | 67 | 65 |
| 250 | 87 | 99 | 85 | 74 |

Таблица 5 – время вычисления кратной точки при w=7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм  Длина k | Бинарный алгоритм | NAF | Window NAF | Sliding window NAF |
| 50 | 17 | 18 | 15 | 14 |
| 75 | 27 | 28 | 23 | 21 |
| 100 | 37 | 37 | 32 | 29 |
| 125 | 47 | 47 | 40 | 38 |
| 150 | 55 | 57 | 48 | 45 |
| 175 | 64 | 68 | 57 | 54 |
| 200 | 74 | 77 | 66 | 61 |
| 250 | 93 | 97 | 82 | 77 |

# Контрольные вопросы

1. Какие основные принципиально различные подходы используются для вычисления кратной точки эллиптической кривой?

Подходы с предвычисленными значениями и без.

1. В чем различие между методами вычисления кратной точки эллиптической кривой для фиксированной точки и для не фиксированной точки?

Для случая фиксированной точки мы можем заранее посчитать некоторую часть точек и хранить их в памяти, чтобы просто обращаться к ним при вычислениях, а не пересчитывать заново.

1. Какие из рассмотренных алгоритмов вычисления кратной точки устойчивы к криптоанализу по побочным каналам?

От атаки по побочным каналам защищены изначально лестница Монтгомери, лестница и комбинированный метод. Для остальных алгоритмов существуют регулярные их версии, которые тоже стойкие к атаке по побочным каналам.

# Вывод

В результате данной работы были изучены различные алгоритмы вычисления кратной точки. Были написаны бинарный алгоритм вычисления, а также версии с NAF. В результате, на небольших значениях k бинарный алгоритм показывает лучшее результаты, по сравнению с NAF. Это связано с тем, что на предвычисления уходит больше времени, чем на сам счет точки.