УДК 004.053

АЛГОРИТМ ФОРМИРОВАНИЯ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ РЕГИСТРА СДВИГА С ЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ С ИСПОЛЬЗОВАНИЕМ ПОРЯДКА ТОЧКИ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Буковшин В.А., Чуб П.А., Черкесова Л.В.

ГОУ ВПО «Донской государственный технический университет», Ростов-на-Дону, e-mail: chia2002@inbox.ru

В рамках данной статьи проводится теоретическое описание формирования псевдослучайного числа на основе порядка точки эллиптической кривой при помощи регистра сдвига с линейной обратной связью. Приводится структурное представление описания необходимых и достаточных условий для создания алгоритма. Предпосылками и одной из важных составляющих рассматриваемой статьи является выбор благоприятных параметров эллиптической кривой, поскольку далее придется, после формирования всех существующих точек эллиптической кривой, определять порядок соответствующей точки. Существенным фактором при формировании псевдослучайного числа служит выбор полинома степени *n*. Чем выше степень многочлена, тем больше случайных значений можно получить. Следовательно, целью данной статьи является разработка алгоритма, способствующего формированию генератора случайных чисел на основе выбранного корректного значения параметров эллиптической кривой. Таким образом, задачи, которые преследуются в данной статье, заключаются в теоретическом описании структуры алгоритма, а также в разработке программного средства, позволяющего получать результирующее число, которое определенным образом составляется на вводимом пользователем многочлене при помощи использования случайно сгенерированных битов последовательности регистров сдвига. Тем самым был реализован удобный интерфейс, позволяющий без особого труда взаимодействовать с программным обеспечением. В качестве результатов работы стоит отметить исследование зависимости формирования псевдослучайных чисел от выбранного полинома. Исследования показали, что при различных многочленах псевдослучайное число генерируется в виде некоторого закона распределения, в качестве которого выступает регистр сдвига с линейной обратной связью.

Ключевые слова: эллиптические кривые, ПСЧ, регистр сдвига, полином, генератор

AN ALGORITHM FOR THE FORMATION OF PSEUDORANDOM NUMBERS BASED ON A SHIFT REGISTER WITH LINEAR FEEDBACK USING THE ELLIPTIC CURVE POINT ORDER

Bukovshin V.A., Chub P.A., Cherkesova L.V.

Don State Technical University, Rostov-on-Don, e-mail: chua2002@inbox.ru

In the framework of this article, a theoretical description of the formation of a pseudo-random number based on the point order of an elliptic curve using a shift register with linear feedback is carried out. A structural representation of the description of the necessary and sufficient conditions for creating an algorithm is given. The prerequisites and one of the important components of this article is the selection of favorable parameters of the elliptic curve, since then, after the formation of all existing points of the elliptic curve, it is necessary to determine the order of the corresponding point. An important factor in the formation of a pseudo-random number is the choice of a polynomial of degree n. The higher the degree of the polynomial, the more random values can be obtained. Therefore, the aim of this article is to develop an algorithm that promotes the formation of a random number generator based on the selected correct value of the parameters of the elliptic curve. Thus, the tasks that are pursued in this article are the theoretical description of the structure of the algorithm, as well as the development of a software tool that allows you to get the resulting number, which is compiled in a certain way on the user input polynomial using randomly generated bits of a sequence of shift registers. Thus, a convenient interface was implemented that allows you to easily interact with the software. As the results of the work, it is worth noting the study of the dependence of the formation of pseudorandom numbers on the selected polynomial. Studies have shown that for various polynomials a pseudo-random number is generated in the form of a certain distribution law, which is a shift register with linear feedback.

Keywords: elliptic curves, pseudo-random numbers, shift register, polynomial, generator

Псевдослучайные числа (ПСЧ) – значения, которые возникают при помощи некоторого закона распределения. Если на мгновение задуматься, то закон распределения – это некоторая математическая функция, которая формирует значение на основе некоторого первоначального числа. Возникает вопрос, откуда взять начальное значение?

На самом деле существует довольно много способов для определения первоначального числа, либо взять максимально возможный диапазон значений и выбрать случайное число, либо при помощи некоторого алгоритма сформировать значение и так далее.

Для конкретной статьи был выбран алгоритм формирования начального чис-

ла с помощью порядка точки эллиптической кривой.

На данный момент существует множество интерпретации в понимании, что такое эллиптическая кривая. На самом деле достаточно будет понимать, что это просто множество точек, описываемое следующим уравнением [1]:

$$y^2 = x^3 + ax + b. \quad (1)$$

В первую очередь при работе с эллиптическими кривыми необходимо исключить особые кривые. То есть нужно прописать проверку, при которой неравенство (2) будет выполняться.

$$4a^3 + 27b^2 > 0. \quad (2)$$

После проверки эллиптической кривой необходимо отметить, что у каждого ненулевого элемента так называемой точки есть либо два корня, либо нет ни одного. Поэтому точки кривой разбиваем на пары следующего вида:

$$P = (x, y) \text{ и } \overline{P} = (x, -y). \quad (3)$$

Важным фактором в эллиптических кривых является то, что любая прямая, проходящая через две различные точки, пересекает данную кривую в третьей точке. Кроме точек перегиба, касательная, проведенная к кривой, пересекает её еще в одной точке. Симметрия кривой относительно Ox позволяет дать наглядное определение обратной к рассматриваемой точке.

Введем две операции, которые можно выполнять над точками [2]:

1) сложение точек – результат суммирования двух точек эллиптической кривой, в конечном итоге которого появляется обратная третья точка;

2) умножение точки на число – прибавление рассматриваемой точки к самой себе K раз.

Разберемся еще с одним понятием, таким как порядок точки кривой – число, при умножении на которое возникает точка на бесконечности.

Так как в дальнейшем подразумевается работа с ПСЧ, необходимо обратить внимание на следующие свойства, которые позволяют определить точки эллиптической кривой над конечными полями:

1) некоторые значения y^2 не имеют квадратного корня по модулю 13;

2) каждая точка имеет инверсию;

3) точки инверсии находятся на тех же самых вертикальных линиях.

После определения эллиптических кривых разберемся с понятием ПСЧ [3]. Важно отметить, что под случайной величиной стоит понимать некоторое значение, полученное в результате опыта. Причем появление той или иной величины нельзя предсказать совершенно точно до её измерения. Поэтому последовательность будет считаться случайной тогда и только тогда, когда следующий шаг никак не зависит от предыдущего.

Исходя из вышеописанного утверждения, формирование ПСЧ основывается на некотором алгоритме, который позволяет в результате получать значения, отличные друг от друга, и подчинен определенному заданному закону распределения.

На практике используются следующие способы генерации случайных чисел (ГСЧ):

– аппаратный, реализация которого не требует дополнительных вычислительных ресурсов ЭВМ по выработке случайных чисел, а необходима только операция обращения к внешнему устройству, так называемому датчику.

– табличный, реализация которого требует определенных затрат в памяти ЭВМ, так как полученные значения записываются в таблицу и хранятся на компьютере;

– алгоритмический, реализация которого при формировании случайных чисел требует больших затрат ресурсов ЭВМ с помощью специальных алгоритмов и реализующих программ.

Для достижения «идеального» ГСЧ необходимо придерживаться следующих правил:

– занимать минимальный объем машинной памяти и времени;

– иметь неповторяющиеся числа;

– быть воспроизводимым;

– содержать статистически независимые числа;

– состоять из квазиравномерно распределенных чисел.

Теперь, после краткого обзора про ПСЧ, разберемся с регистром сдвига с линейной обратной связью [4].

Данный регистр состоит из двух частей:

– регистр сдвига;

– функция обратной связи.

Пример регистра сдвига с линейной обратной связью (РСЛОС) представлен на рис. 1.

Рис. 1. Регистр сдвига с линейной обратной связью

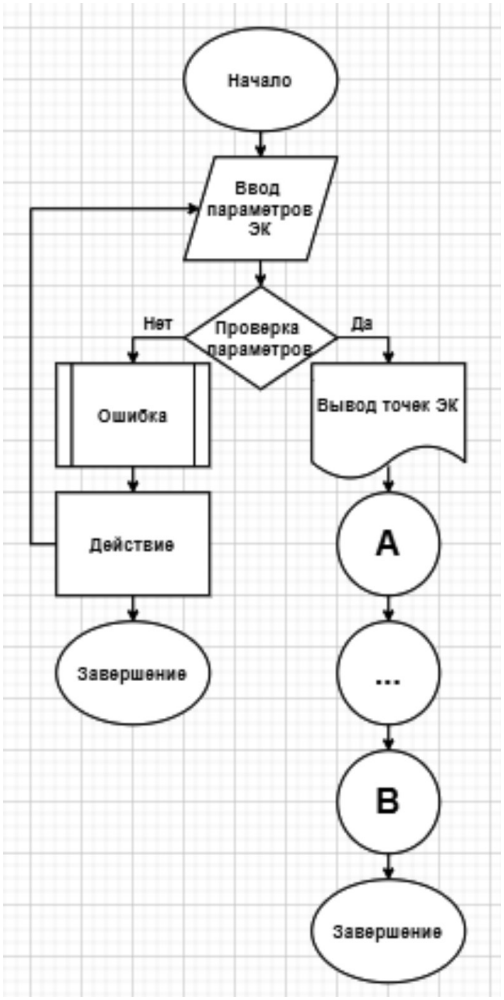

Рис. 2. Блок-схема программы (ч. 1)

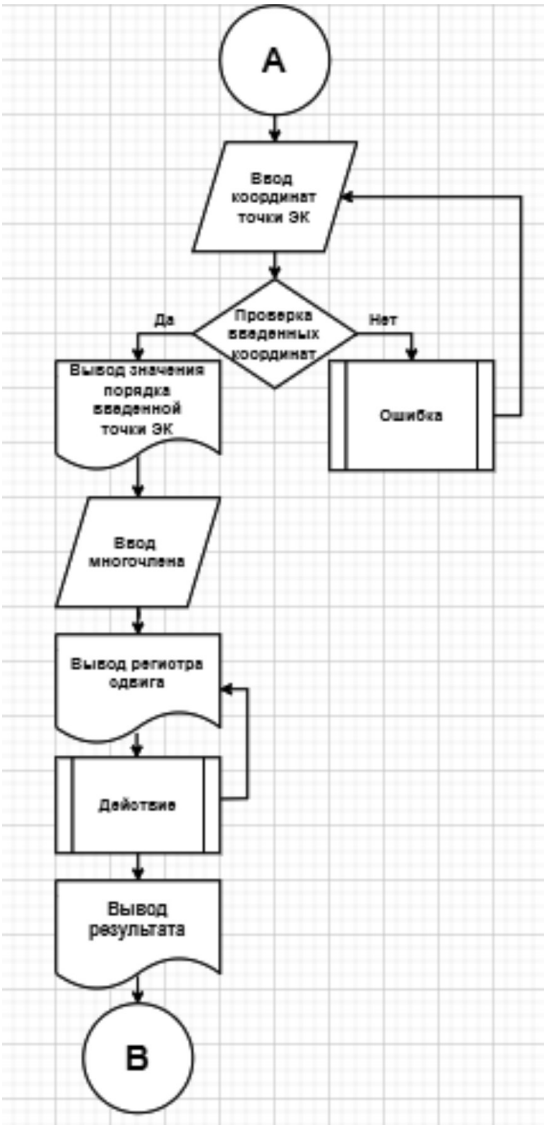

Рис. 3. Блок-схема программы (ч. 2)

Сам регистр сдвига представляет собой последовательность бит, число которых определяет длину регистра. При каждом извлечении бита все биты сдвигаются вправо на одну позицию в сторону младших разрядов.

В качестве обратной связи выступает любая математическая функция, которая производит действие над битами. В рамках статьи будет использоваться XOR-преобразование случайных бит. Запись в РСЛОС производится в виде полиномов. Важно отметить, что степень первого полинома указывает на количество битов в регистре сдвига, а степенные показатели остальных элементов полинома указывают, какие будут использованы ячейки регистра сдвига при съеме битов для математических операций.

Обобщая полученные утверждения, можно сказать, что *n*-битовый регистр сдвига с линейной обратной связью может находиться в одном из $2^n - 1$ внутренних состояний.

Рассмотренный регистр может генерировать ПСП с использованием эллиптических кривых над конечными полями.

Распишем пошагово алгоритм формирования ПСЧ:

– выбираем конечное поле;

– выбираем ЭК, в которой не будут присутствовать исключения;

– выбираем точку ЭК порядка K;

– выбираем примитивный многочлен для формирования регистра сдвига;

– вычисляем последовательность P с использованием M-последовательности q с выходом конечного элемента;

– преобразуем полученное значение в двоичную последовательность с использованием оператора отображения;

– после преобразуем целое число из двоичной последовательности [5].

Исходя из вышеописанного алгоритма, был разработан программный продукт, при помощи которого можно получить ПСЧ или ГСП с использованием порядка выбранной точки, принадлежащей ЭК. На блок-схеме, представленной на рис. 2–3, структурно показано тело программы.

Опишем более детально функционал программного средства.

Для удобства работы с программой был разработан интерфейс, представленный на рис. 4 [6].

На начальном этапе пользователю необходимо ввести параметры эллиптической кривой. В качестве проверки введем следующие параметры: E_{211} (5, -4). Данное действие продемонстрировано на рис. 5. Выбранное поле должно быть простым числом, а параметр «a» – взаимно простым с конечным полем.

Прежде чем продолжить работу с программой, нужно удостовериться в правильности введенных параметров. Для этого была разработана проверка, чтобы исключить возможность ошибочного ввода данных. Описанное действие выполняет специальная кнопка, представленная на рис. 6.

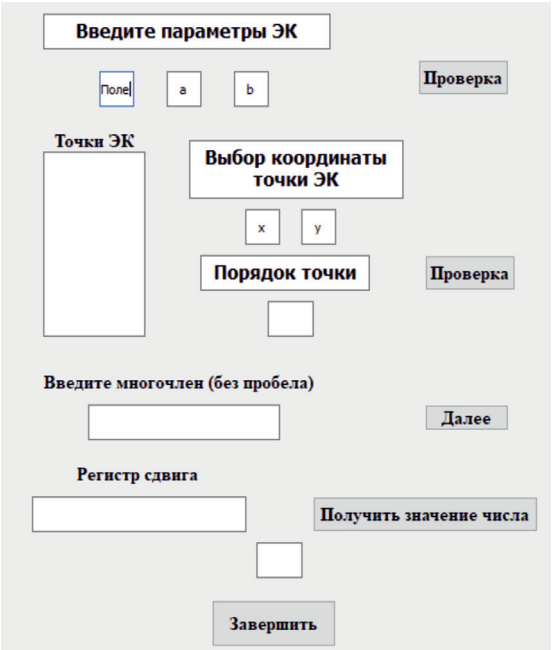

Рис. 4. Визуальное окно программы

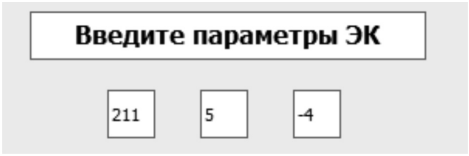

Рис. 5. Окно ввода данных

Рис. 6. Кнопка, отвечающая за проверку введенных данных

После проверки введенных параметров, если данные – корректны, то в том случае в окне, показанном на рис. 7, будет выведено множество точек, принадлежащих рассматриваемой эллиптической кривой.

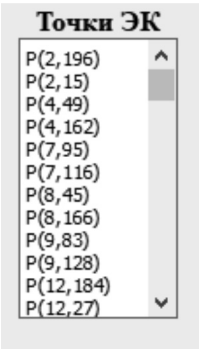

Рис. 7. Окно вывода точек ЭК

Исходя из рис. 7, можно утверждать, что введенные пользователем параметры – корректны, следовательно, в окне ввода координат точки ЭК, представленной на рис. 8, выбирается точка, которая принадлежит множеству точек эллиптической кривой. В виде исключения была написана проверка ввода корректных данных пользователем. За описанное действие отвечает кнопка, показанная на рис. 9.

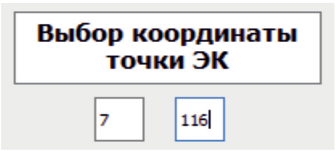

Рис. 8. Окно ввода координат точки ЭК

Рис. 9. Кнопка проверки корректности ввода данных

В том случае, если проверка успешно пройдена, определяется порядок выбранной точки, как видно из рис. 10.

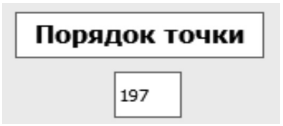

Рис. 10. Окно вывода порядка точки ЭК

После вывода числа пользователю предоставляется возможность для ввода многочлена вида

$$xn + x(n-1) + \cdots + x(n-(n-1)) + 1,$$

где n – максимальная степень многочлена и длина регистра сдвига.

В качестве проверки на рис. 11 продемонстрирован многочлен.

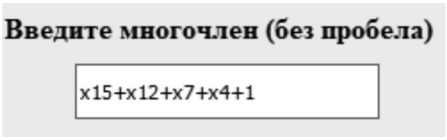

Рис. 11. Окно ввода полинома

После нажатия на кнопку далее, как показано на рис. 12, при помощи преобразований получаем регистр сдвига. Окно вывода представлено на рис. 13.

Рис. 12. Кнопка для преобразования регистра сдвига

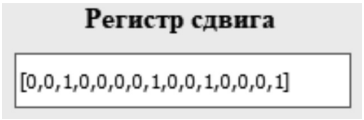

Рис. 13. Окно вывода сдвигового регистра

Затем, при нажатии на кнопку, показанную на рис. 14, формируется ПСЧ на основе полученного регистра сдвига, путем выбранной случайной последовательности бит регистра. Полученный результат отображается в окне вывода, представленном на рис. 15.

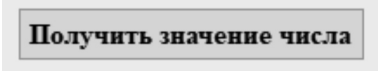

Рис. 14. Формирование ПСЧ

Рис. 15. Окно вывода результата

Если необходимо вывести сгенерированную последовательность без повторяющихся значений, а не одно, то необходимо нажимать на кнопку, показанную на рис. 14, до тех пор, пока не выведется окно вывода, которое продемонстрировано на рис. 16, и затем, нажав на кнопку, показанную на рис. 17, можно просмотреть сгенерированную последовательность, результат которой приведен на рис. 18. После окончания работы с программным средством необходимо нажать на кнопку, показанную на рис. 19.

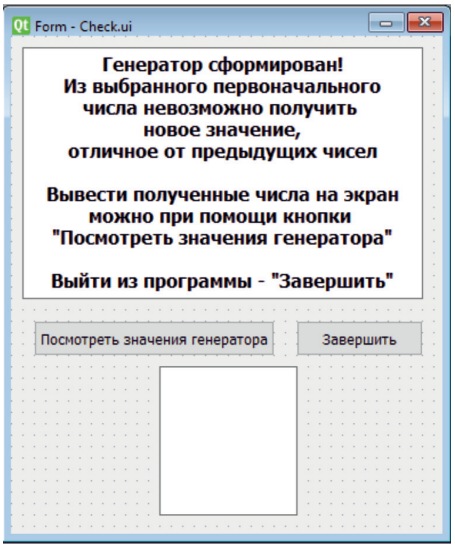

Рис. 16. Окно вывода

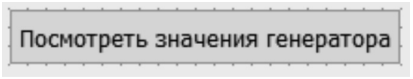

Рис. 17. Кнопка, отвечающая за вывод генератора

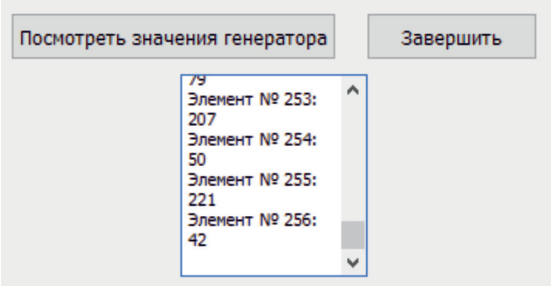

Рис. 18. Фрагмент полученной последовательности

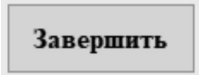

Рис. 19. Завершение работы программы

Теперь опишем функционал программного средства.

На рис. 20 можно наблюдать функцию, которая отвечает за проверку введенных пользователем параметров ЭК. Изначально проверяется на простоту. Затем, после прохождения условия, проверяется параметр a. Если рассматриваемый параметр имеет отрицательное значение, то к нему прибавляется значение поля, для исключения особых кривых. После необходимо, чтобы наибольший общий делитель поля, относительно параметра a, был равен 1.

Как можно увидеть на рис. 20, после прохождения всех условий, есть некая функция, которая называется result. Рассматриваемая функция отвечает за генерацию точек ЭК. Это можно наблюдать на рис. 21. В этой функции проверяется кривая на наличие особых кривых, когда значение дискриминанта меньше либо равно 0. Затем, если рассматриваемая ЭК не является особой, то генерируются точки ЭК и выводятся на экран пользователю.

На рис. 22 изображена функция, в которой используются введенные пользователем значения точки ЭК, в результате чего формируется вывод порядка точки, нажав на кнопку, которая представлена на рис. 9.

После определения порядка точки, на рис. 23–24 представлены фрагменты программного кода, которые отвечают за ввод пользователем полинома и вывода полученного регистра на экран, в результате нажатия на кнопку, которая представлена на рис. 12.

```
def psh_Btn(self):
    field, par_a, par_b = self.value()
    bool = checking.test_Ferma(field)
    if bool == False: self.check_Error()
    else:
        if par_a < 0: par_a+=field
        if par_a == 0: self.result(par_a, par_b, field)
        else:
            if checking.check_NOD(par_a, field) != 1: self.check_Error()
            else: self.result(par_a, par_b, field)
```

Рис. 20. Проверка введенных параметров ЭК

```
def result(self, a, b, field):

    bool_shit = checking.check_elleptic(a, b, field)
    if bool_shit == False:
        self.check_Error()
    else:
        list_x, list_y, count = point_definition.forming_point(a, b, field)
        str_1 = point_definition.output_point(list_x, list_y)
        print(str_1)
        self.textBrowser_7.setText(str_1)
```

Рис. 21. Функция генерации точек ЭК

```
def psh_Btn_1(self):
    field, par_a, par_b = self.value()
    val_coord_x, val_coord_y = self.input_value()
    list_all_coord_x, list_all_coord_y, count = point_definition.forming_point(par_a, par_b, field)
    bool = checking.check_point(val_coord_x, val_coord_y, list_all_coord_x, list_all_coord_y)
    if bool == False:
        self.check_Error_2()
    else:
        val = checking.point_order(count, val_coord_x, val_coord_y, par_a, field)
        self.generator_numbers.append(val)
        self.lineEdit_4.setText(str(val))
```

Рис. 22. Определение порядка точки ЭК

```
def psh_Btn_2(self):
    val = self.lineEdit_7.text()
    self.lineEdit_8.setText(work_with_poly.poly(val))
```

Рис. 23. Фрагмент кода, отвечающий за вывод регистра сдвига на экран

```
def poly(val):
    buff = []
    while len(val) != 0:
        ind = get_length_reg(val)
        if ind == len(val)-1:
            if val != '1': buff.append(int(val[1:]))
            else: buff.append(0)
            break
        else:
            buff.append(int(val[1:ind]))
            val = val[(ind+1):]
    length_reg = buff[0]
    buff.remove(buff[0])
    reg = shift_reg(buff, length_reg)
    return conv_str_2(reg)
```

Рис. 24. Программный код, в результате которого формируется регистр сдвига

```
def psh_Btn_3(self):
    val = int(self.lineEdit_4.text())
    print(self.generator_numbers)
    self.result_1(val, self.generator_numbers)
```

Рис. 25. Функция вывода значения на экран

```
def result_1(self, val, buff_lst):
    val_1 = self.lineEdit_8.text()
    reg = work_with_poly.form_reg(val_1)
    self.lineEdit_9.setText(str(self.return_val(buff_lst, work_with_poly.bin_to_int(val, reg), reg)))

def return_val(self, generator_numbers, val, shift_register):

    while 1:
        if checking.check_elem(generator_numbers, val) == False:
            if self.count == len(generator_numbers):
                self.window = Check_1(generator_numbers)
                self.window.show()
                self.close()
                break
            else:
                val = work_with_poly.bin_to_int(generator_numbers[0], shift_register)
                print('Value parameter is count for number not input in list: ', self.count)
                self.count += 1
        else:
            generator_numbers.append(val)
            self.count = 1
            print('Count value in list: ', len(generator_numbers), '\nValue parameter is count: ', self.count)
            return val
```

Рис. 26. Фрагмент кода, отвечающий за формирование ПСЧ

Из рис. 25 можно просмотреть функцию, которая по начальному значению, а именно по порядку точки ЭК, выводит каждое последующее уникальное значение ПСЧ, которое основано на регистре сдвига с линейной обратной связью, нажав на кнопку, показанную на рис. 14.

Затем за получение ПСЧ, или сгенерированной последовательности, отвечает функция, которая представлена на рис. 26.

Рассматриваемая функция включает в себя проверку полученного значения с теми, которые уже присутствуют в сгенерированной последовательности. Если данное число присутствует, то анализируется последующее полученное число, в результате воздействия регистра сдвига. Если уже все числа – конечны и невозможно получить уникальное значение, которое может включать в себя генератор ПСЧ, то выводится сообщение, как показано на рис. 16, в результате чего можно просмотреть сгенерированную последовательность.

И, наконец, на рис. 27 проиллюстрирована функция, отвечающая за завершение работы программы.

```
def psh_Btn_4(self):
    sys.exit()
```

Рис. 27. Функция завершения работы программы

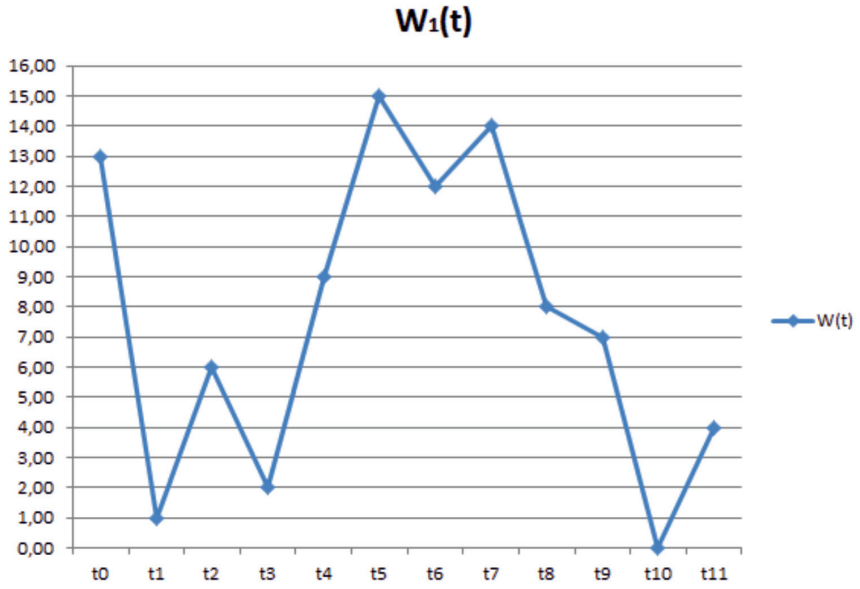

Рис. 28. График зависимости ПСЧ от времени

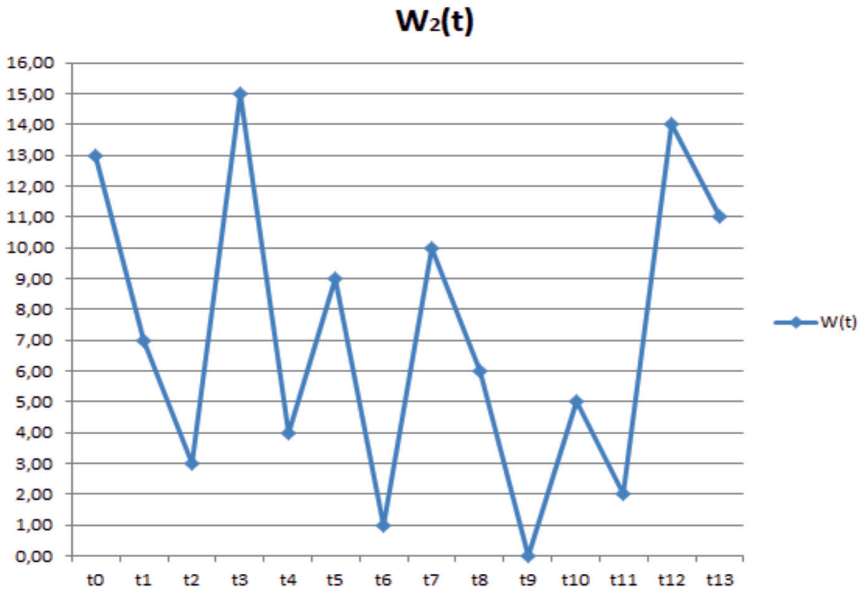

Рис. 29. График зависимости ПСЧ от времени

Исследование показателей сгенерированных ПСЧ относительно выбранного многочлена

В рамках эксперимента для улучшения наглядности результатов значения параметров ЭК выбирались одинаковыми. Стоит принять во внимание тот факт, что в основе формирования генератора ПСЧ на основе порядка точки ЭК лежит многочлен. Таким образом, в качестве исследования выбирались несколько различных полиномов.

На рис. 28 приведена зависимость полученного ПСЧ относительно временного промежутка, то есть в определенный момент после нажатия на кнопку, показанную на рис. 14, генерировалось случайное число. Выбирается полином вида: $x^{10} + x^7 + x^5 + x + 1$ и при помощи программного средства формируется генератор ПСЧ.

В качестве W выступает последовательность случайных чисел, а t – временной промежуток.

На рис. 29 представлена зависимость ПСЧ от времени, но отличие состоит в том, что многочлен имеет следующий вид: $x^5 + x^4 + x^3 + x$.

Таким образом, можно сделать вывод, что при различных полиномах ПСЧ генерируются в определенный момент времени по некоторому закону распределения, основанного на первоначальном значении, в качестве которого выбирается порядок точки ЭК.

В виде закона распределения выступает регистр сдвига с линейной обратной связью.

Заключение

Программа, которая была разработана в ходе статьи, позволяет:

1) определять корректность введенных значений;

2) удобно работать пользователю;

3) формировать ПСЧ на основе первоначального значения в виде порядка точки и регистра сдвига с линейной обратной связью;

4) проводить исследования при формировании генератора ПСЧ.

Список литературы

1. Жданов О.Н., Чалкин В.А. Эллиптические кривые. Основы теории и криптографические приложения: лабораторный практикум. М.: URSS, 2013. 200 с.

2. Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А. Элементарное введение в эллиптическую криптографию // Алгебраические и алгоритмические основы: учебное пособие. М., 2019. 376 с.

3. Слеповичев И.И. Генераторы псевдослучайных чисел: учебное пособие. М.: Академия, 2017. 118 с.

4. Хорев П.Б. Методы и средства защиты информации в компьютерных системах. М.: Академия, 2007. 256 с.

5. Иванов М.А., Чугунков И.В. Криптографические методы защиты информации в компьютерных системах и сетях: учебное пособие. М., 2012. 400 с.

6. Дональд Кнут. Искусство программирования. Т. 2. Получисленные алгоритмы. М.: «Вильямс», 2007. 832 с.