УДК 519

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ГЕНЕРАТОРА ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ АЛГОРИТМА БЛЮМ – БЛЮМА – ШУБА

А. Е. Каргин
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Изучены и проанализированы основные принципы работы генератора псевдослучайных чисел. Описан генератор псевдослучайных чисел на основе алгоритма Блюм — Блюма — Шуба. Разработано программное средство для генерации псевдослучайных чисел на основе алгоритма Блюм — Блюма — Шуба. Результаты работы программы представлены на снимках и проанализированы.

Ключевые слова: криптография, криптостойкость, псевдослучайное число, генератор псевдослучайных чисел, алгоритм Блюм — Блюма — Шуба.

SOFTWARE IMPLEMENTATION OF THE PSEUDORANDOM NUMBER GENERATOR ON THE BLUM – BLUM – SHUB ALGORITHM

A. E. Kargin
Don State Technical University (Rostov-on-Don, Russian Federation)

The article studies and analyzes the basic principles of the pseudorandom number generator. The Blum — Blum — Shub pseudorandom number generator is described. The Pseudorandom number generator on the Blum — Blum — Shub algorithm has been developed. The results of the program are presented in the pictures and analyzed.

Keywords: cryptography, cryptographic strength, pseudorandom number, pseudorandom number generator, Blum — Blum — Shub algorithm.

Введение. На сегодняшний день задача генерации случайных чисел является актуальной во многих сферах жизнедеятельности человека — от индустрии развлечений и социологических исследований до компьютерного моделирования и информационной безопасности. Особенно остро стоит проблема в криптографических системах. Для большинства прикладных задач, таких как поточное шифрование или получение уникальных значений, используется генерация псевдослучайных чисел [1].

В начале XX века случайные числа генерировали путем бросания монеты, извлечением шаров из урны или с помощью рулетки. Позже Леонардом Типпетом была опубликована таблица со случайными значениями. А в 1939 году для создания таблицы, которая содержала в себе 100 000 случайных чисел, был использован аппаратный генератор. Но ни табличный метод, ни аппаратные генераторы не давали эффективных и надежных результатов. В связи с этим в 1946 году американский математик Джон фон Нейман предложил арифметический способ генерации случайной последовательности. Он назывался «Методом серединных квадратов» и суть его заключалась в следующем: для получения очередного числа необходимо возвести предыдущее в квадрат и выделить из него цифры посередине [2]. Полученные в результате числа не были случайными, ведь они вычислялись с помощью определенной формулы, хотя и были похожи на случайную последовательность. Такие числа называют псевдослучайными, а метод их генерации — генератором псевдослучайных чисел (ГПСЧ).

Обычный генератор псевдослучайных чисел несовершенен. В ходе своей работы он зацикливается — начинает повторять одну и ту же последовательность значений. Также к недостаткам можно отнести неравномерное распределение и отсутствие независимости последовательных значений.

Современные криптографические системы используют псевдослучайные числа для получения ключей или набора входных параметров алгоритма шифрования. Для таких целей необходим криптостойкий генератор псевдослучайных чисел (КСГПСЧ), который удовлетворяет следующим требованиям: он должен проходить статистические тесты на случайность, а также сохранять непредсказуемость, даже если часть его текущего состояния становится известна криптоаналитику [3]. Помимо этого, криптографически стойкий генератор должен выполнять те же требования, которые предъявляются и к обычному генератору. Одним из таких КСГПСЧ является алгоритм Блюм – Блюма – Шуба.

Алгоритм Блюм – Блюма – Шуба, предложенный в 1986 году Ленором Блюмом, Мануэлем Блюмом и Майклом Шубом, на сегодняшний день является одним из самых быстрых и простых алгоритмов. Основу его работы составляет вычисление квадратичных остатков по модулю M [4].

Целью статьи является программная реализация генератора псевдослучайных чисел на основе алгоритма Блюм – Блюма – Шуба.

Основная часть. Процесс генерации последовательности из n псевдослучайных бит состоит в следующем:

– генерируются два больших простых числа p и q, они должны быть сравнимы с 3 по модулю 4;

– число M, называемое целым числом Блюма, вычисляется как произведение чисел p и q;

– выбирается случайное целое число x, взаимно простое с M;

– стартовое число генератора находится по следующей формуле:

$$x_0 = x^2 mod\ M \quad (1)$$

– вычисляем последующие целые числа по формуле:

$$x_n = x_{n-1}^2 mod\ M \quad (2)$$

– из каждого n-го числа в его двоичном представлении выбирается младший бит.

В результате получаем n псевдослучайных бит.

Например, нужно получить последовательность из 5 бит. Пусть p = 359, q = 463. Данные числа простые и сравнимы с 3 по модулю 4. Вычисляем M = 359 × 463 = 166217. Выбираем взаимно простое с M число x = 5. С помощью формулы (1) вычисляется стартовое число x_0 = 25. Далее по формуле (2) вычислим следующие 5 чисел x_i и запишем в таблицу (1).

Таблица 1

Результаты вычислений x_i

i	**x_{i-1}**	**x_i**	**младший бит**
1	25	625	1
2	625	58191	1
3	58191	19757	1
4	19757	61533	1
5	61533	53046	0

Результатом работы генератора стала последовательность пяти псевдослучайных бит: 11110.

Одним из преимуществ данного алгоритма является получение i-го числа без вычисления предыдущих значений. Для этого можно воспользоваться следующей формулой:

$$x_i = x_0^{2^i mod((p-1)(q-1))} mod\ M \quad (3)$$

Имея значения q, p и x_0, вычислим x_3:

$x_5 = 25^{2^3 mod((359-1)(463-1))} mod\ 166217 = 25^{8 mod 165396} mod\ 166217 = 152587890625 mod 166217 = 19757.$

Использовав формулу (3), получили число 19757, такое же, как и при последовательном вычислении.

Данное свойство полезно при потоковом шифровании, в частности, при работе с массивами данных с произвольной точкой доступа.

Надежность алгоритма Блюм — Блюма — Шуба основывается на вычислительной сложности задачи факторизации числа M. Также данный алгоритм генерирует последовательность псевдослучайных бит с большим периодом, что дает возможность применять его при генерации ключей [5]. Исходя из этого, алгоритм Блюм — Блюма — Шуба можно назвать криптографически стойким генератором псевдослучайных чисел.

Программная реализация. В качестве основного инструмента для разработки программы был выбрал язык программирования Python [6], ключевыми достоинствами которого являются:

– автоматическая работа с освобождением памяти;

– кроссплатформенность;

– поддержка всех кодировок;

– большое количество встроенных математических библиотек;

– высокая читаемость кода.

Генерация начальных параметров p и q выполняется основе вероятностного теста простоты Рабина-Миллера. Для поддержки высокой криптостойкости алгоритма длина каждого из параметров p и q будет составлять 512 бит.

Подробный алгоритм работы, в котором происходит генерация последовательности значений для дальнейшего получения псевдослучайных бит, показан на рис. 1. Сгенерированное число имеет длину, равную 10 бит.

Из примера выше видно, что полученная последовательность псевдослучайных бит преобразуется в десятичное число, которое и является результатом работы генератора.

На рис. 2 представлен пример генерации 30 псевдослучайных чисел, размер каждого из которых составляет 100 бит. Происходит последовательная генерация 3000 значений от x_1 до x_{3000}, каждый из которых имеет длину до 1024 бит. Из каждого x_i вычисляется бит четности. Затем формируются 30 списков из 100 бит. Далее происходит преобразование каждого набора бит в десятичное число. В результате на выходе получаем 30 псевдослучайных чисел.

При использовании одних и тех же входных параметров ГПСЧ будет генерировать одинаковую последовательность чисел, что показано на рис. 3. Это подтверждает детерминированность алгоритма. Данное свойство удовлетворяет одному из качественных требований, предъявляемых к генератору псевдослучайных чисел, — воспроизводимости — возможности в любое время заново сгенерировать последовательность значений необходимое количество раз. Обновление генератора, реализованное в программе, представляет из себя сброс очередного числа x_i и присвоение ему значения x_0, сгенерированного в начале работы алгоритма.

```
Генерация псевдослучайного числа размером 10 бит.

Начальные параметры:
p = 12252188843008021995581629185027523673015972450996758314196474893358288574192538292586838740725177469348073927007405784203611547190368727072608069372939639
q = 82273198223693627701040764655253566167820187788402581022145557797751397989906572883316597010111799296679964911834393978586797503025864896364425722595947 9
M = 10080267613549264788015243685926909282215263837689750257726217132802670030159549048171272593140015139313163091912680346407288283325478069415208335470747748399288797777230626394374819548930742498140470081936806565201472203159328430093225892600599314363474049648178210047044562396632862858673012435945928898808 1
x0 = 43142758593784769735774067601377854452302754718760302074056106793475871757525363866740508467530849955145854431234160854310376450653736824516955868150911615717509679968117458374880453181439264481760277318153548270211762809465062428304576408494980563611549457619286485089797042241480089915860282739104881789708

x1: 29873086024081093808526538735204189335187853922188682346155033214525615758580076973932269188925924230794625394630210857871460810465195789374127081867869537441929296792177399539996969831988306940282544920271784480408739650938743811049607685887597916679063145635482620619539921903220699237466954194148205190137
x2: 57689809033438981160353358182378784981377890843198835631571482791941930507017385630279596420662335353901210642467005827614740405863513392788433738089424008391716316496649092002216006138352250250267325725971588912068465304888480970692719868492874926296083665458681620369904865566339801443479449253003657622717
x3: 68861851743181741113135060873019903883116782758125936596272773497225732079529662703093906141548188065411846070108216268707298782103317500984874860298200998985632988846886577547112014665441509838626066116214131489299717933072704686273898125716941951630457461312347115816850278266204824251738877134345715705673
x4: 61793679431795922462023539874003758996017274309447866000275921105448848977924067395503804961896566749045609635694968572940464779389182315008256122773444173504089082776263333058516647073935769040925687668716245948667003333811245175276214392524012762914853996198836061793862168486508039263663039822066926725548
x5: 55201220608952556631103653012920356395517385715359322882738203503290544825413310765319942582782633155993966099416850618840745008191202498470491404399351171113319122633317663479640692538489494118337301276819436243884780338715299007990609200283392952789023582155659653258540756541565223844095201157322088709068
x6: 61250655008370747023710337018267581027227305248202494974932123600968982049903597513455840259693932175969700439822600425313918948526534041754551167209320922812341853195373877554052836026029188883717303225437101007599481632377281881596272634753896303463227638509677948559816181032190957627591667279756459953 19
x7: 17056052089070370541078314776606374341525124906602002000234721636391910869275353272739781081838002708467566599118248700026402110160685397952219560902127693945397943311737618766987089334549652337311240100704568144081524679437899850588658021048496759217371214474583426584035479062475926456075544611167589045323
x8: 58788740832519956173755020105583839846003261761756226681390986499368130259131766199243076896895527031371582017757019215348414213433462602604300015513252560795838648042234449480308029243366228017827682728261796918024581194503418586560950170090477661479718329
x9: 19321177093593299811958876395454550909320072384651698977686482803075866938308973930024051999484103711051175742279177551496408327690201052093659100621930699353526262763343354724023729484652140000861480498393458339397491996514698643965127755716696156887243163303768780930987676164158955592330986793545205767 7
x10: 16041207010620006035657586914520108670724090861996983201335036119674714309142924705827665880283871696172128061092950818351148916331476256663207778404909390991577375118363273949766698725692865376975387389350189335158198086496056359344286556227537190388794582118692381592974982048121105751228117151054652535579
Полученные биты: 1110011111

Сгенерированное число: 927
```

Рис. 1. Подробный пример генерации псевдослучайного числа

```
Генерация 30 псевдослучайных чисел размером по 100 бит:

1-е число = 6185934933488740923663241708 33
2-е число = 98137181590833330781304391848
3-е число = 46907517223579271973977747303
4-е число = 33367736597891450139508319349 1
5-е число = 10519702447750728666566866112858
6-е число = 56918619184914173927901134757 3
7-е число = 11389151478121822935722555601464
8-е число = 52014033546123210906123608713 7
9-е число = 12018354465220082100318176701 45
10-е число = 15542156265979015805232742374 2
11-е число = 18611327177689487787976066551 2
12-е число = 53811461596723652469403007789 9
13-е число = 11382948836023585110890394963 87
14-е число = 84443074609250884274937144050 1
15-е число = 90092915239743265233313244352 6
16-е число = 55135486859439505246577724573 2
17-е число = 18489162004970923082063765786 7
18-е число = 11662170690182707513038291439 47
19-е число = 11263252551255027283627463848 16
20-е число = 25278601073840127892109795423
21-е число = 93268668745466821087182436063
22-е число = 67635153369400425843612968970 7
23-е число = 32097707049118388802363595493 5
24-е число = 99490509680653308346878747041 6
25-е число = 73648865556647275028662790645 7
26-е число = 76287700118209263940182588831 4
27-е число = 16998187076022651978905771536 1
28-е число = 42426517895746666989082572389
29-е число = 64769827118135727232998530054 9
30-е число = 79943402943849429196682199239 2
```

Рис. 2. Пример генерации 30 псевдослучайных чисел

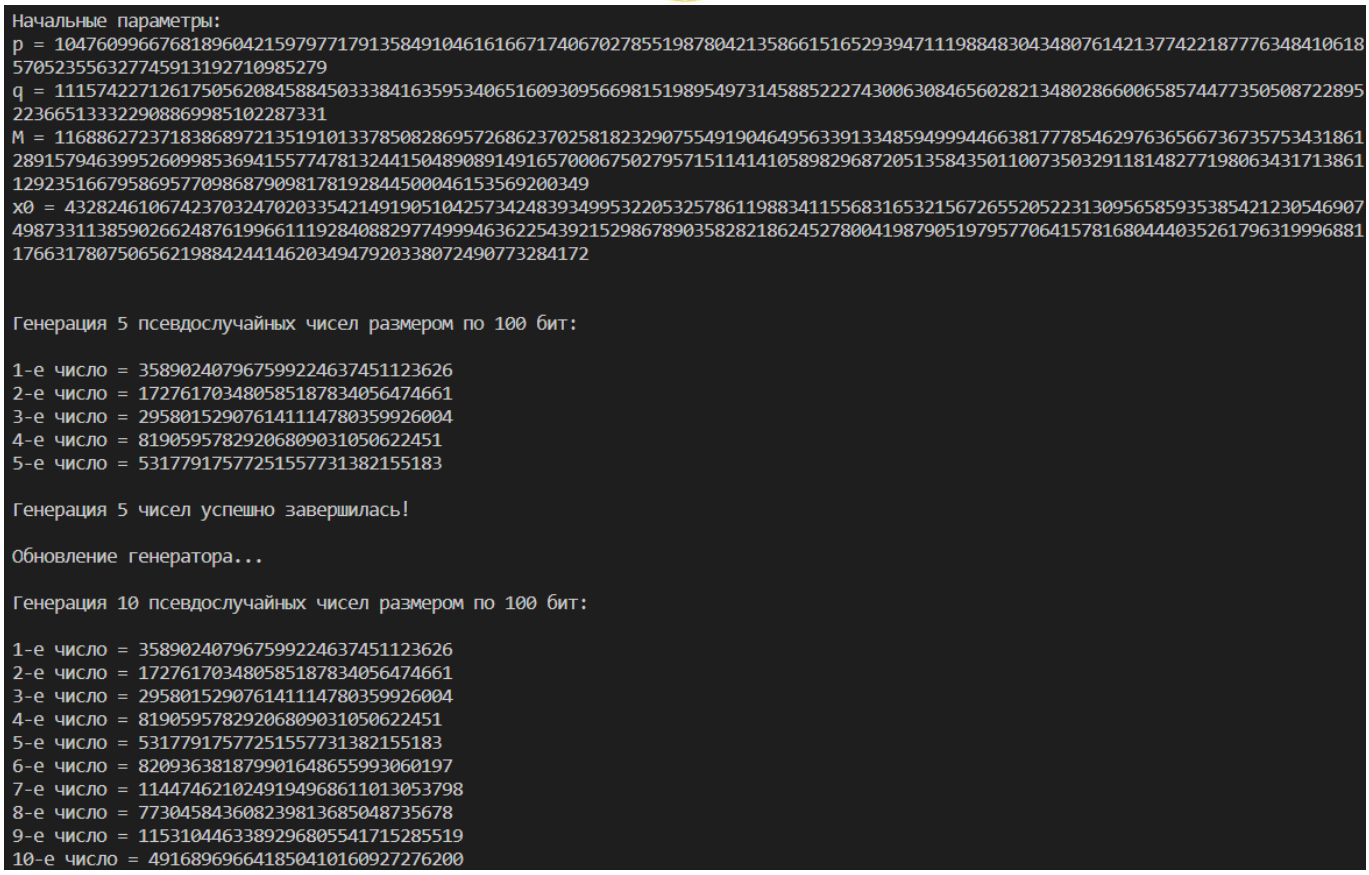

Рис. 3. Генерация чисел с одинаковыми входными данными

Заключение. В результате проделанной работы были изучены и проанализированы основные принципы работы генератора псевдослучайных чисел. Разработано программное средство для генерации псевдослучайных чисел на основе алгоритма Блюм — Блюма — Шуба. Программное средство способно генерировать последовательность из больших значений, удовлетворяя основным требованиям к криптографически стойким генераторам псевдослучайных чисел, поэтому данный алгоритм может применяться на практике в криптографии. Таким образом, все поставленные задачи выполнены, а цели достигнуты.

Библиографический список

1. Слеповичев, И. И. Генераторы псевдослучайных чисел / И. И. Слеповичев. — Саратов : СГУ, 2017. — 118 с. — URL : https://www.sgu.ru/sites/default/files/textdocsfiles/2018/07/09/slepovichev_i.i._generatory_psevdosluchaynyh_chisel_2017.pdf (дата обращения : 20.03.2022).

2. Генератор псевдослучайных чисел / Википедия :[сайт]. — URL : https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB (дата обращения : 20.03.2022).

3. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы и исходные такты на языке Си / Б. Шнайер — Москва : Триумф, 2002. — 816 с.

4. Алгоритм Блюм – Блюма – Шуба / Академик :[сайт]. — URL : https://dic.academic.ru/dic.nsf/ruwiki/614129 (дата обращения : 20.03.2022).

5. Выборнова, Ю. Д. Разработка функции генерации ключа на основе пароля в качестве приложения генератора Blum-Blum-Shub / Ю. Д. Выборнова // Информационные технологии и нанотехнологии (ИТНТ-2017) : сб. тр. III междунар. конф и молодежной школы — Самара: Предприятие «Новая техника», 2017. — С. 888–895.

6. Официальный сайт языка программирования Python / Python : [сайт]. — URL : https://www.python.org/ (дата обращения : 20.03.2022).

Об авторе:

Каргин Артем Евгеньевич, студент кафедры «Компьютерная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), cargin.art@yandex.ru

About the Author:

Kargin, Artem E., Student of the Computer Security Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), cargin.art@yandex.ru.