Министерство образования и науки Российской Федерации

ФГБОУ «Петрозаводский государственный университет»

Институт математики и информационных технологий

Отчет по лабораторной работе

Генерация совместного ключа методом Диффи-Хеллмана

**Выполнили:**

**студент 3 курса 22305 Д. И. Кнауб**

**студентка 3 курса 22305 Д. С. Ефимова**

**Преподаватель по лабораторной**

**работе: д.т.н., доцент,**

**профессор кафедры ПМиК**

**Р. В. Воронов**

подпись принявшего работу

**Петрозаводск – 2022**

**Оглавление**

[**Формулировка задания** 3](#_Toc98543821)

[**Описание метода решения** 4](#_Toc98543822)

[Генерация параметров 4](#_Toc98543823)

[Тест Рабина-Миллера 5](#_Toc98543824)

[Возведение в степень по модулю в кольце 6](#_Toc98543825)

[Надежные числа 6](#_Toc98543826)

[Создание общего секретного ключа 6](#_Toc98543827)

[Имитация обмена данными между двумя пользователями 7](#_Toc98543828)

[**Примеры кода программы** 8](#_Toc98543829)

[**Тестовые данные** 12](#_Toc98543830)

# **Формулировка задания**

**Вариант 2.6.**

Напишите программу генерации совместного ключа методом Диффи-Хеллмана. Рекомендуется использовать библиотеку для работы с длинными числами. В случае применения этой библиотеки разрешается использовать функции сложения, вычитания, умножения, целочисленного деления, вычисления остатка от деления. Функции возведения числа в степень, нахождения наибольшего общего делителя, обратного элемента в мультипликативной группе вычетов, генерации простого числа реализовать самостоятельно. Использовать «надежные» простые числа. Выполняемые функции программы:

1) генерация чисел p, g, x, y;

2) имитация обмена данными между пользователями;

3) получение общего ключа.

# **Описание метода решения**

Одна из фундаментальных проблем криптографии – безопасное общение по прослушиваемому каналу. Сообщения нужно зашифровывать и расшифровывать, но для этого обеим сторонам нужно иметь общий ключ. Если этот ключ передавать по тому же каналу, то прослушивающая сторона тоже получит его, и смысл шифрования исчезнет.

**Метод Диффи-Хеллмана** позволяет двум сторонам получить общий секретный ключ, используя незащищенный от прослушивания, но защищённый от подмены канал связи. Полученный ключ можно использовать для обмена сообщениями с помощью симметричного шифрования.

Предположим, существует два пользователя: П1 – первый пользователь и П2 – второй пользователь. Рассмотрим работу данного метода между этими пользователями.

## Генерация параметров

Для реализации метода необходимы два параметра: p – большое простое число, g – примитивный элемент, порождающий мультипликативную группу вычетов по модулю р . Мультипликативная группа вычетов – группа, состоящая из вычетов , взаимно простых с р. Групповой операцией является умножение по модулю p: , а примитивный элемент (или первообразный корень) – это такой элемент, что все его степени по модулю p пробегают по всем p – 1 числам, взаимно простых с p. Число g также является большим и простым числом.

Числа p, g открыты и могут использоваться несколькими пользователями одновременно. П1 и П2 должны заранее договориться о данных числах для дальнейшего их использования (обычно значения генерируются одной стороной и передаются другой).

На самом деле параметр g может и не быть примитивным элементом, а порождал подгруппу достаточно большого порядка. Поэтому генерация параметров p, g будет производиться следующим образом (Листинг 1):

1. Выбираем в качестве q большое простое и надежное число.
2. Находим первый параметр , для некоторого четного n. Это число подбирается случайно до тех пор, пока р не будет простым.
3. Переходим к поиску второго параметра g порядка q. Выбирается случайное число a из , которое возводится в степень n по модулю p: . Если g = 1, то повторяем предыдущий шаг. Найденное значение g будет иметь порядок q, при этом такое g порождает подгруппу, размер которой является делителем q.

После генерации параметров их необходимо проверить (Листинг 2):

1. q является делителем р – 1
2. g ≠ 1, , g < p
3. p, q – простые, надежные и длинные числа

Если данные условия выполняются, то получившиеся параметры можно использовать для составления общего ключа.

В методе решения используется следующий подход к генерации больших простых чисел (Листинг 4): случайным образом генерируется большое целое число, величина которого задана количеством бит, а затем выполняется тест этого числа на простоту, который представляет из себя вероятностный метод теста Рабина-Миллера. Так как по условию задачи необходимо использовать надежные числа, то, если число прошло тест Рабина-Миллера, оно дальше отправляется на проверку надежности (Листинг 3).

Тест Рабина-Миллера (Листинг 5) позволяет эффективно определять, является ли данное число составным, однако, с его помощью нельзя строго доказать простоту числа. На вход тесту подается большое целое число n, которое каждый раз генерируется функцией random.getrandbits(Amount\_of\_bits) до тех пор, пока не найдется простое число, а также подается число итераций k. Каждая итерация Рабина-Миллера уменьшает вероятность того, что число является составным в ¼ раза. Оптимальным количеством итераций является число 40, которое и используется в алгоритме. Сразу же проверяем: если число четное, то оно составное.

Число n – 1 однозначно представляется в виде Это можно сделать последовательным делением n – 1 на 2.

Далее повторяем цикл k раз: выбирается случайное целое число а в отрезке [2, n – 1]. Данное число называется свидетелем простоты числа n, если:

* , то продолжаем генерировать новое число а, пока не пройдем все k проверок. x вычисляется с помощью возведения в степень по модулю.

Иначе, повторить цикл s – 1 раз (внутренний цикл): если , то перейти на следующую итерацию внешнего цикла. Иначе вернуть False.

Если все k итераций пройдены, то вернуть True (простое).

Обычно при реализации необходимо возводить число в очень большую степень по модулю, что при использовании функции python pow(a,b,c) занимает большое количество времени. Поэтому была реализована функция возведения в степень по модулю в кольце (Листинг 6).

Возведение в степень по модулю в кольце выглядит следующим образом: пока степень p больше квадрата, то

* Если степень нечетная, то получение четной степени происходит за счет выноса множителя, перемножая с уже имеющимся и находя остаток при делении этого произведения. Сама степень p понижается на 1.
* Если степень четная, то понижаем ее, возводя число в квадрат и находя остаток от деления, а степень уменьшаем в 2 раза. Это возможно сделать, так как по модулю есть большое количество эквивалентных чисел данному.

В завершении (когда степень сократится до квадрата или меньше) оставшийся элемент, возведенный в получившуюся степень по модулю, перемножается с полученным результатом, поделенное на модуль.

### Надежные числа

В криптографии с открытым ключом часто применяются надежные (безопасные) простые числа, называемые числами Софи-Жерман. Они важны в криптографии в виду их использования в подходах, основанных на дискретных логарифмах, таких как алгоритм Диффи-Хеллмана. Если 2p + 1 безопасное простое, мультипликативная группа чисел по модулю 2p + 1 имеет подгруппу высокого порядка. Обычно эта та подгруппа, которая нужна, и причина использования безопасных простых чисел заключается в том, что модуль мал относительно p.

Итак, надежное число – это такое просто число y, что число 2y+1 также простое.

Когда требуется сгенерировать простое число, оно проходит проверку на надежность (Листинг 3). Генератор простых чисел генерирует простое число y, после чего оно уходит на проверку «надёжности», если число 2y+1 является простым, то число y «надёжное» и его можно использовать для дальнейших вычислений.

## Создание общего секретного ключа

Для создания общего секретного ключа П1 и П2 должны выполнить следующие шаги (Листинг 7):

1. П1 выбирает случайное, никому более неизвестное число а, которое также проходит проверку на надежность, а, значит, является простым.
2. П2 также выбирает случайное, никому более неизвестное число b, которое также проходит проверку на надежность, а, значит, является простым.
3. П1 вычисляет открытый ключ
4. П2 вычисляет открытый ключ
5. П1 и П2 обмениваются открытыми ключами А и В.
6. П1 вычисляет свой секретный ключ
7. П2 вычисляет свой секретный ключ

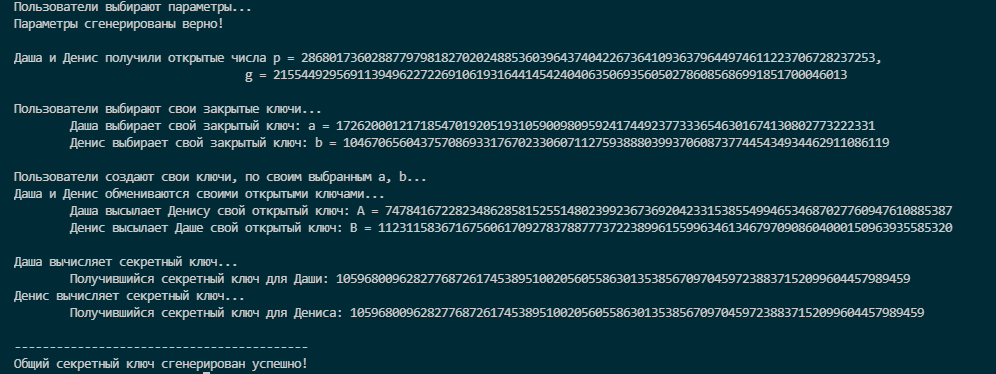
Если все шаги выполнены верно, то

Причем при выполнении 3, 4 шагов, для сокращения времени выполнения операции, используется вычисление , что сокращает скорость вычислений в 8 раз (Листинг 8). Также необходимо выполнить проверку, что А, В принадлежат группе, порожденной элементов g. Для этого должно выполняться равенство: .

Надежность метода обусловлена трудностью задачи дискретного логарифмирования. При неизвестных a, b быстро вычислить не получится.

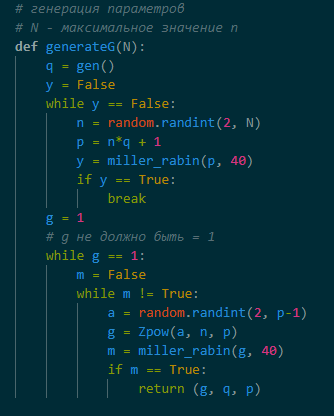
Имитация обмена данными между двумя пользователями показана на Рисунке 1. Благодаря такой имитации, можно наблюдать за действиями каждого из двух пользователей.

Рисунок 1. Имитация обмена данными

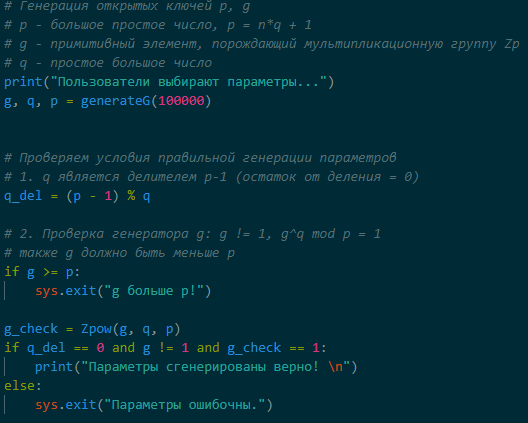


# **Примеры кода программы**

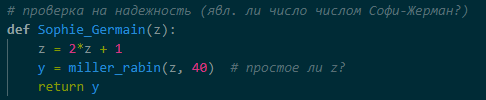
Листинг 1. Генерация параметров p, g, q



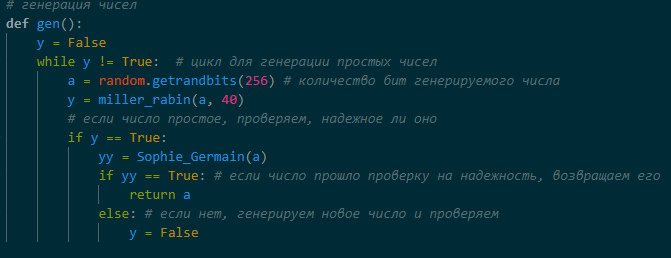
Листинг 2. Проверка сгенерированных параметров



Листинг 3. Проверка на надежность простых чисел



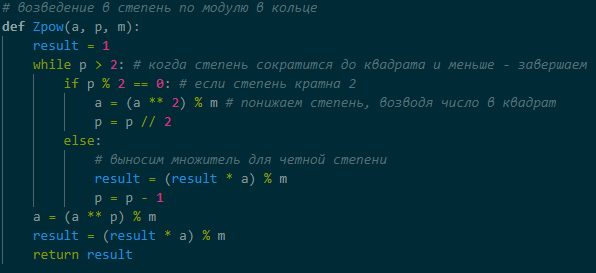
Листинг 4. Генерация больших чисел



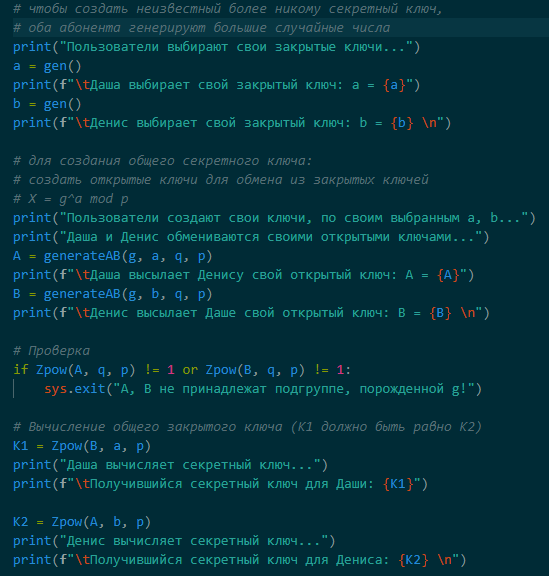
Листинг 5. Тест Рабина-Миллера на простоту чисел

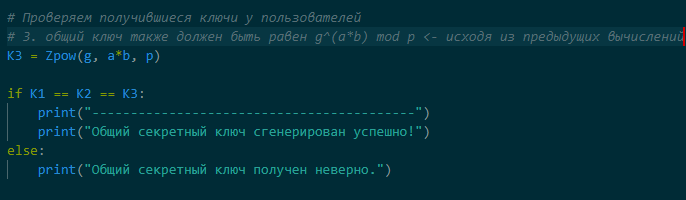


Листинг 6. Функция возведения в степень по модулю в кольце

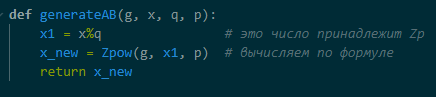


Листинг 7. Создание общего секретного ключа





Листинг 8. Генерация открытых ключей А и В



# **Тестовые данные**

Проверка работы алгоритма проводилась вручную на тестовых данных (результаты 10 тестов приведены ниже). Для проверки также было засечено время выполнения программы. Проанализировав полученные результаты, можно сказать, что алгоритм работает безошибочно, но при увеличении значений генерируемых чисел, время работы программы заметно замедляется. Отметим, что работа замедляется именно при генерации чисел, а не при вычислениях. Это связано из-за огромного количества проверок чисел на простоту и надежность.

Тест 1.

|  |  |
| --- | --- |
| p | 1187 |
| g | 673 |
| a | 2 |
| b | 1019 |
| A | 682 |
| B | 635 |
| K1 | 832 |
| K2 | 832 |
| Время выполнения | 0.003553499999725318 |

Тест 2.

|  |  |
| --- | --- |
| p | 28814050884506539 |
| g | 2327600159485033 |
| a | 308368289057981 |
| b | 746035997761361 |
| A | 3180321235143821 |
| B | 20399057502389515 |
| K1 | 18820213754222982 |
| K2 | 18820213754222982 |
| Время выполнения | 0.11304170000039448 |

Тест 3.

|  |  |
| --- | --- |
| p | 7398696472034075143834813 |
| g | 5601959479276362184480073 |
| a | 208593943751547587131631 |
| b | 286234517443599515589083 |
| A | 3336521169684066606367998 |
| B | 241879785398872350818346 |
| K1 | 1189561024873857274911719 |
| K2 | 1189561024873857274911719 |
| Время выполнения | 0.46261219999996683 |

Тест 4.

|  |  |
| --- | --- |
| p | 8515165754446936335093283117742079928591 |
| g | 1734475613852528426884537032254780212679 |
| a | 228666775821027041118231442965404539409 |
| b | 112381164265269520538690582341675105013 |
| A | 6295216901172996780260619181015731021567 |
| B | 5750584599525684992407074508172511326728 |
| K1 | 2272397270740435352059025422162525785674 |
| K2 | 2272397270740435352059025422162525785674 |
| Время выполнения | 3.44682019999982 |

Тест 5.

|  |  |
| --- | --- |
| p | 120818340076012326416404393380662950140391 |
| g | 106089630995858328665630752765482745373123 |
| a | 102353051785633002383738177101865926709 |
| b | 174805580170421111812271461364588305853 |
| A | 41311501560807236155594708890671544322327 |
| B | 33559000293124241285306479774528887465056 |
| K1 | 57147607239259234596422450191782079501272 |
| K2 | 57147607239259234596422450191782079501272 |
| Время выполнения | 2.854569000000083 |

Тест 6.

|  |  |
| --- | --- |
| p | 4062005380061454659383340379223250589944834881192730381 |
| g | 3483771135245124397965382495418771444633599258270544707 |
| a | 1075662155258855664962260986726956424985598125282049 |
| b | 1242902885843209513752888352185481196791481126481779 |
| A | 1348256970691201614670057951629536611727191016227946442 |
| B | 2112590483624710681836527676953831888883571302124005539 |
| K1 | 3586460839337210345651856422900845462694474399759451869 |
| K2 | 3586460839337210345651856422900845462694474399759451869 |
| Время выполнения | 2.2601886000002196 |

Тест 7.

|  |  |
| --- | --- |
| p | 1046538942307834690651532059850797769460950036290374668521405991 |
| g | 650227055057319417136326040903678691858496518171800511576317231 |
| a | 759441526525458338430982991481983107256378656001599728738431 |
| b | 1287227765308242800065781932741542716741185158489983088682919 |
| A | 84265935470605766110631579240504474082619914105878227007145787 |
| B | 558887874385982835279455871429854761730270055769279813352207781 |
| K1 | 224318856139644463852743072275646328448535556133149231890415141 |
| K2 | 224318856139644463852743072275646328448535556133149231890415141 |
| Время выполнения | 7.2558861999996225 |

Тест 8.

|  |  |
| --- | --- |
| p | 202161646053931105102707461526389171497272041920872290880663661179980785056952261 |
| g | 26421504159779614284062848324947945599805495398275346448956153358460223883976303 |
| a | 25263011524992148532755121385245532783741261528825029106094011448020241912433 |
| b | 106086856973616586849826473056070850385504359193120555870610886707543423264361 |
| A | 15087932017073269254806990534654791159113292057144214354688507017853976015072715 |
| B | 43634077748357012231382932376767580385978213097500867583553591804766465323564795 |
| K1 | 69765904486900295279007669590518213630194278809170214458679227457579441134902445 |
| K2 | 69765904486900295279007669590518213630194278809170214458679227457579441134902445 |
| Время выполнения | 23.974428600000465 |

Тест 9.

|  |  |
| --- | --- |
| p | 390121723980273644395753663525115514314114520902023335788701233520515780150933161 |
| g | 1334404400922188709683950438659262433558934914508614951056437927637426117676197 |
| a | 35197822550784206158378302367146020902040692295985979458924148850622208031813 |
| b | 8835874889766619893623058342727089513141296920442922389923372402106178943019 |
| A | 224381285811051798179132573795367464800821116145905800555194352698477086478485242 |
| B | 329746358450289125146491279880103746091733292822583456727438579041828762032634630 |
| K1 | 346246455253890731980069340488614350277667235872059377400029556391280859316999420 |
| K2 | 346246455253890731980069340488614350277667235872059377400029556391280859316999420 |
| Время выполнения | 16.18482059999951 |

Тест 10.

|  |  |
| --- | --- |
| p | 186472337270713997849989843728796244023922362775385526281060025831106458208128400494025221151471263688768113104340265485864935531216585496319787842185452594353 |
| g | 114735014923399922651279749080747753592678853641077581461125697507381810392678298844524860577841091419295851652681288221569122651545357070440954467960587258211 |
| a | 778867260223778133391370536052974055333974561695405551089303136982009047904344329438060718708544600607480419593781613883774367360505487160383446124558183 |
| b | 2329415040627005151633306470732814280760471828490692376554371162773279287270054934181609181785592196757926209266050933320499519272357836719146827041376829 |
| A | 176778951751137134203801393852211894250088164696378057831752331186621659016560893745036033274303339382085163053605885457140752709464510302082503211872410279932 |
| B | 109082383378703871391624766176918914610579484888794710921913902862206978517961010156144789527242044511868516897727381904238620310418296440291969391024723564949 |
| K1 | 104661653450463613907142160122341804432332189874269877608457680123461541412477040787879920086612459478097943945732316355547966724434732189028570895863133652388 |
| K2 | 104661653450463613907142160122341804432332189874269877608457680123461541412477040787879920086612459478097943945732316355547966724434732189028570895863133652388 |
| Время выполнения | 428.45438860000013 |