Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Саратовский государственный университет имени Н.Г.Чернышевского»

Балашовский институт (филиал)

С. А. Бубнов

Лабораторный практикум по основам криптографии

Учебно-методическое пособие
для студентов по профилю подготовки 080801.65 «Прикладная информатика (в экономике)»

Саратов 2012

УДК 004
ББК 32.97я73
Б 90

Учебно-методическое пособие преподавателя кафедры прикладной информатики Балашовского института Саратовского университета кандидата физико-математических наук Бубнова Сергея Алексеевича предназначено для студентов по профилю подготовки 080801.65 «Прикладная информатика (в экономике)». Все задания лабораторных работ без особых трудностей реализуются в среде программирования Delphi, с основами которой студенты уже знакомы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ..... 4

1. Лабораторная работа №1..... 5
2. Лабораторная работа №2..... 8
3. Лабораторная работа №3..... 12
4. Лабораторная работа №4 15
5. Лабораторная работа №5 19
6. Лабораторная работа №6 24
7. Лабораторная работа №7 27
8. Лабораторная работа №8 31

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА..... 34

ВВЕДЕНИЕ

Криптографические методы защиты информации используются человечеством с давних времен. Потребность в них возникла сразу после становления письменности и государства. Ярким примером первой «шифровальной машины» можно считать сциталь – деревянный цилиндр, на который наматывался пояс виток к витку. На этом поясе вдоль оси сциталя записывалось сообщение. После разматывания пояса сообщение невозможно было прочитать. Ключом в данной системе является диаметр сциталя. Существовали и более сложные модели – вместо цилиндра применялся конус.

Другим историческим примером системы шифрования является шифр замены, в котором буквы исходного текста заменяются на символы шифралфавита. Например, так шифровали во времена правления Цезаря.

Криптография – это первая составляющая более общей науки, называемой криптологией. Криптография занимается построением отображений информации (криптографических отображений), применяемых с целью ее защиты, т.е. построением шифров. Криптоанализ является второй составляющей криптологии и занимается методами анализа криптографических отображений с целью раскрытия защищаемой информации.

На первых этапах развития криптология не располагала строгими математическими доказательствами применяемых методов и идей. Все основывалось на интуитивном уровне исходя из практических соображений. В настоящее время криптология является серьезной наукой, включающей в себя такие разделы дискретной математики, как множества и отображения, функции алгебры логики, функции k-значной логики, сбалансированность отображений, структура и периоды преобразований, псевдослучайные последовательности, шифрующие автоматы.

ЛАБОРАТОРНАЯ РАБОТА №1

Шифры простой замены

1. Шифр Цезаря.

В основе функционирования шифров простой замены лежит следующий принцип: для получения шифртекста отдельные символы или группы символов исходного алфавита заменяются символами или группами символов шифроалфавита.

Шифр Цезаря (также он является шифром простой замены) – это моноалфавитная подстановка, т.е. каждой букве открытого текста ставится в соответствие одна буква шифртекста. На практике при создании шифра простой замены в качестве шифроалфавита берется исходный алфавит, но с нарушенным порядком букв (*алфавитная перестановка*). Для запоминания нового порядка букв перемешивание алфавита осуществляется с помощью пароля. В качестве пароля могут выступать слово или несколько слов с неповторяющимися буквами. Шифровальная таблица состоит из двух строк: в первой записывается стандартный алфавит открытого текста, во второй – начиная с некоторой позиции размещается пароль (пробелы опускаются), а далее идут в алфавитном порядке оставшиеся буквы, не вошедшие в пароль. В случае несовпадения начала пароля с началом строки процесс после ее завершения циклически продолжается с первой позиции. Ключом шифра служит пароль вместе с числом, указывающим положение начальной буквы пароля. Таблица шифрования на ключе *4 пароль* будет иметь вид:

а б в г д е ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я

ы э ю я **п а р о л ь** б в г д е ж з и й к м н с т у ф х ц ч ш щ ъ

В процессе шифрования каждая буква открытого текста заменяется на стоящую под ней букву.

В 1 в. н.э. Ю. Цезарь во время войны с галлами, переписываясь со своими друзьями в Риме, заменял в сообщении первую букву латинского алфавита (A) на четвертую (D), вторую (B) – на пятую (E), наконец, последнюю – на третью:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Донесение Ю. Цезаря Сенату об одержанной им победе над Понтийским царем выглядело так:

YHQL YLGL YLFL ("Veni, vidi, vici" – лат. "Пришел, увидел, победил").

Император Август (1 в. н. э.) в своей переписке заменял первую букву на вторую, вторую – на третью и т. д., наконец, последнюю – на первую:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

B C D E F G H I J K L M N O P Q R S T U V W X Y Z A

Любимое изречение императора Августа выглядело так:

GFTUJOB MFOUF ("Festina lente" – лат. "Торопись медленно").

Из примеров видно, что изменяя величину сдвига, можно получить несколько разных криптограмм для одного исходного текста.

Математически процедуру шифрования можно описать следующим образом:

$$T_m = \{T^j\}, j = 0,1, \dots, m-1,$$

$$T^j(a) = (a+j) mod\ m,$$

где $(a+j) mod\ m$ – операция нахождения остатка от целочисленного деления $a+j$ на m; T_m – циклическая подгруппа. Пронумеруем буквы латинского алфавита от 0 до 25: $a = 0, b = 1, c = 3, \dots, z = 25$. В латинском алфавите 26 букв и поэтому примем $m = 26$. Тогда операцию шифрования запишем в виде: буква с номером i заменяется на букву с номером $(i+3)\ mod\ 26$. Возможно и обобщение шифра Цезаря на случай произвольного ключа k: символ с номером i заменится на символ с номером $(i+k) mod\ 26$.

Таким образом, открытый текст $a_0, a_1, \dots, a_{N-1}$ преобразуется в криптограмму $T^j(a_0), T^j(a_1), \dots, T^j(a_{N-1})$. При использовании для шифрования подстановки T^j символ a открытого текста заменяется символом $a+j$

шифрованного текста. Цезарь обычно для шифрования использовал подстановку T^3.

Взлом такого шифра осуществляется путем анализа частотных характеристик языка открытых текстов. Например, в русском тексте длиной 10000 символов буква О встречается в среднем 1047 раз, Е – 836, А – 808, Н – 723 и т.д. Поэтому, если в достаточно длинной криптограмме какой-то символ встречается чаще остальных, то есть все основания полагать, что это буква О.

2. Шифр Атбаш.

Данный шифр является шифром сдвига на всю длину алфавита. Для алфавита, состоящего только из русских букв и пробела, таблица шифрования будет иметь следующий вид:

а б в г д е ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я ␣

␣ я ю э ь ы ъ щ ш ч ц х ф у т с р п о н м л к й и з ж е д г в б а

При программной реализации шифра Атбаш на языке Pascal целесообразно использовать таблицу ASCII и функции работы с ней (ord и char). Далее показана функция перевода символа открытого текста в шифр путем зеркального отражения по таблице ASCII.

```
Function Atbash(openchar:char):char;
 Begin
  Atbash := 255 – ord(openchar);
End.
```

Задания к лабораторной работе

1. Реализовать шифр Цезаря с произвольным ключом k.
2. Реализовать шифр Атбаш.

ЛАБОРАТОРНАЯ РАБОТА №2

Шифры перестановки

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов. Способ, каким при шифровании переставляются буквы открытого текста, и является ключом шифра. Важным требованием является равенство длин ключа и исходного текста.

Существует два широко распространенных метода перестановок:

1. **Маршрутное шифрование.**

Данный способ шифрования разработал французский математик Франсуа Виет. Открытый текст записывают в некоторую геометрическую фигуру (обычно прямоугольник) по некоторому пути, а затем, выписывая символы по другому пути, получают шифртекст. Пусть m и n – целые положительные числа, большие 1. Открытый текст разбивается на блоки равной длины, состоящие из числа символов, равному произведению mn. Если последний блок получится меньше остальных, то в него следует дописать требуемое количество произвольных символов. Составляется таблица размерности mn. Блоки вписывается построчно в таблицу. Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Ключом такой криптограммы является маршрут и числа m и n. Обычно буквы выписывают по столбцам, которые упорядочивают согласно паролю: внизу таблицы приписывается слово из n неповторяющихся букв и столбцы нумеруются по алфавитному порядку букв пароля.

Например, для шифрования текста *нельзя недооценивать противника*, разобьем его на блоки длины $n = 6$. Блоков получится $m = 5$. К последнему блоку припишем букву *а*. В качестве пароля выберем слово *пароль.* Теперь будем выписывать буквы по столбцам в соответствии с алфавитным порядком букв пароля и получим следующую криптограмму: ЕЕНПНЗОАТАЬОВОКННЕЬВЯЦТИА.

н	*е*	*л*	*ь*	*з*	*я*
н	*е*	*д*	*о*	*о*	*ц*
е	*н*	*и*	*в*	*а*	*т*
ь	*п*	*р*	*о*	*т*	*и*
в	*н*	*и*	*к*	*а*	*а*
п	*а*	*р*	*о*	*л*	*ь*

Рассмотренный способ шифрования (столбцовая перестановка) в годы первой мировой войны использовала легендарная немецкая шпионка Мата Хари.

2. Шифрование с помощью решеток.

Данный способ шифрования предложил австрийский криптограф Эдуард Флейснер в 1881 году. Суть этого способа заключается в следующем. Выбирается натуральное число $k > 1$, строится квадрат размерности k и построчно заполняется числами $1, 2, \dots, k^2$. В качестве примера рассмотри квадрат размерности $k = 2$.

1	2
3	4

Повернем его по часовой стрелке на 90^0 и присоединим к исходному квадрату справа.

1	2	3	1
3	4	4	2

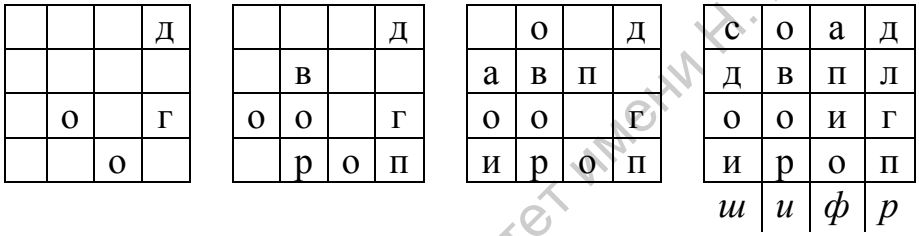
Проделаем еще дважды такую процедуру и припишем получившиеся квадраты снизу. Получился большой квадрат размерности $2k$.

1	2	3	1
3	4	4	2
2	4	4	3
1	3	2	1

Далее из большого квадрата вырезаются клетки, содержащие числа от 1 до k^2. В каждой клетке должно быть только одно число. Получается своего рода решето. Шифрование осуществляется следующим образом. Решето накладывается на чистый квадрат $2k \times 2k$ и в прорези вписываются буквы

исходного текста по порядку их следования. Когда заполнятся все прорези, решето поворачивается на 90^0 и вписывание букв продолжается. После третьего поворота все клетки большого квадрата окажутся заполненными. Подобрав подходящий пароль (число букв пароля должно равняться k^2 и они не должны повторяться), выпишем буквы по столбцам. Очередность столбцов определяется алфавитным порядком букв пароля.

Пример. Исходный текст – *договор подписали*; пароль – *шифр*. С применением вышеуказанной решетки за пять шагов получаем следующую криптограмму.



Получившаяся криптограмма: ОВОРДЛГПАПИОСДОИ. Важно отметить, что число k подбирается в соответствии с количеством букв N исходного теста. В идеальном случае $k^2 = N$. Если такого равенства достичь невозможно, то можно либо дописать произвольную букву к последнему слову открытого текста, либо убрать ее.

3. Таблица Виженера.

В 1585 году французский криптограф Блез Виженер опубликовал свой метод шифрования в «Трактате о шифрах». Шифр считался нераскрываемым до 1863 года, когда австриец Фридрих Казиски взломал его.

Открытый текст разбивается на блоки длины n. Ключ представляет собой последовательность из n натуральных чисел: $a_1, a_2, \dots, a_n$. Далее в каждом блоке первая буква циклически сдвигается вправо по алфавиту на a_1 позиций, вторая буква – на a_2 позиций, последняя – на a_n позиций. Для лучшего запоминания в качестве ключа можно взять осмысленное слово, а алфавитные номера входящих в него букв использовать для осуществления сдвигов. Рассмотрим еще одну

схему построения шифра Виженера. В нижеприведенной таблице в строчках записаны буквы русского алфавита. При переходе от одной строке к другой происходит циклический сдвиг на одну позицию. Исходный текст: *криптография серьезная наука;* пароль – *математика*. Пароль записывается с повторениями над буквами сообщения.

м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а
к	р	и	п	т	о	г	р	а	ф	и	я	с	е	р	ь	е	з	н	а	я	н	а	у	к	а

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я
Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А
В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б
Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В
Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г
Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д
Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е
З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь
Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы
Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э
Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю

В горизонтальном алфавите находим букву «к», а в вертикальном – букву «м». На пересечении столбца и строки в таблице расположена буква «ц». Далее переходим к буквам «р» и «а» соответственно. В итоге получается следующая криптограмма: ЦРЬФЯОХШКФФЯДКЭЬЧПЧАЛНТШЦА.

Задания к лабораторной работе

Реализовать все рассмотренные шифры программно.

ЛАБОРАТОРНАЯ РАБОТА №3

Шифрование гаммированием

Из всех схем шифрования простейшей и наиболее надежной является схема однократного использования (рис. 1). Формируется m- разрядная случайная двоичная последовательность – ключ шифра. Отправитель производит побитовое сложение по модулю два ($mod\ 2$) ключа

$$k = k_1 k_2 \dots k_i \dots k_m$$

и m- разрядной двоичной последовательности

$$p = p_1 p_2 \dots p_i \dots p_m,$$

соответствующей посылаемому сообщению:

$$c_i = p_i \oplus k_i, i = \overline{1, m},$$

где p_i – i-й бит исходного текста, k_i – i-й бит ключа, $\oplus$ – операция побитового сложения (XOR), c_i – i-й бит получившейся криптограммы

$$c = c_1 c_2 \dots c_i \dots c_m.$$

Операция побитного сложения является обратимой, т.е. $(x \oplus y) \oplus y = x$, поэтому дешифрование осуществляется повторным применением операции $\oplus$ к криптограмме:

$$p_i = c_i \oplus k_i, i = \overline{1, m}.$$

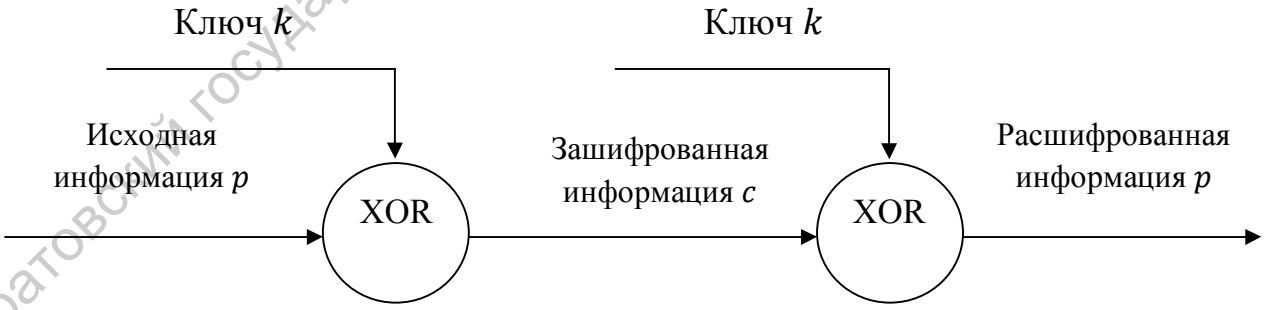


Рис. 1

Основным недостатком такой схемы является равенство объема ключевой информации и суммарного объема передаваемых сообщений. Данный недостаток можно убрать, использовав ключ в качестве «зародыша», порождающего

значительно более длинную ключевую последовательность. На рис. 2. представлена такая схема, которая и называется *гаммированием.*

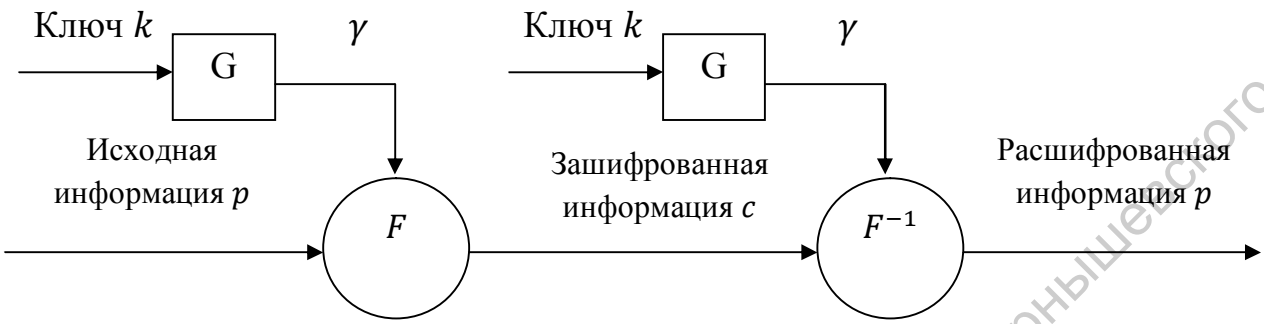


Рис. 2.

Гаммирование – процедура наложения при помощи некоторой функции F на исходный текст *гаммы* шифра, т.е. *псевдослучайной последовательности (ПСП)* с выходов генератора G. Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, т.е. известен алгоритм ее формирования. Чаще Обычно в качестве функции F берется операция поразрядного сложения по модулю два или по модулю N (N – число букв алфавита открытого текста).

Простейший генератор псевдослучайной последовательности можно представить рекуррентным соотношением:

$$\gamma_i = a \cdot \gamma_{i-1} + b \; mod(m), i = \overline{1, m},$$

где γ_i – i-й член последовательности псевдослучайных чисел, a, γ_0, b – ключевые параметры. Такая последовательность состоит из целых чисел от 0 до $m - 1$. Если элементы γ_i и γ_j совпадут, то совпадут и последующие участки: γ_{+1i}=γ_{j+1}, $\gamma_{i+2} = \gamma_{j+2}$. Таким образом, ПСП является периодической. Знание периода гаммы существенно облегчает криптоанализ. Максимальная длина периода равна m. Для ее достижения необходимо удовлетворить следующим условиям:

1. b и m – взаимно простые числа;
2. $a - 1$ делится на любой простой делитель числа m;
3. $a - 1$ кратно 4, если m кратно 4.

Стойкость шифров, основанных на процедуре гаммирования, зависит от характеристик гаммы – длины и равномерности распределения вероятностей появления знаков гаммы.

При использовании генератора ПСП получаем бесконечную гамму. Однако, возможен режим шифрования конечной гаммы. В роли конечной гаммы может выступать фраза. Как и ранее, используется алфавитный порядок букв, т.е. буква «а» имеет порядковый номер 1, «б» – 2 и т.д.

Например, зашифруем слово «ПРИКАЗ» («16 17 09 11 01 08») гаммой «ГАММА» («04 01 13 13 01»). Будем использовать операцию побитового сложения по модулю 33 ($mod\ 33$). Получаем:

$c_1 = 16 + 4(mod\ 33) = 20$ $\quad$ $c_4 = 11 + 13(mod\ 33) = 24$

$c_2 = 17 + 1(mod\ 33) = 18$ $\quad$ $c_5 = 1 + 1(mod\ 33) = 2$

$c_3 = 9 + 13(mod\ 33) = 22$ $\quad$ $c_6 = 8 + 4(mod\ 33) = 12.$

Криптограмма: «УСХЧБЛ» («20 18 22 24 02 12»).

Задания к лабораторной работе

Реализовать алгоритм шифрования гаммированием конечной гаммой.

ЛАБОРАТОРНАЯ РАБОТА №4

Вычисление наибольшего общего делителя

Пусть числа a и b целые и $b \neq 0$. Разделить a на b с остатком – значит представить a в виде $a = qb + r$, где $q, r \in Z$ и $0 \leq r \leq |b|$. Число q называется неполным частным, число r – неполным остатком от деления a на b.

Целое число $d \neq 0$ называется *наибольшим общим делителем* целых чисел $a_1, a_2, \ldots, a_k$ (обозначается $d = \text{НОД}(a_1, a_2, \ldots, a_k)$), если выполняются следующие условия:

1. каждое из чисел $a_1, a_2, \ldots, a_k$ делится на d;
2. если $d_1 \neq 0$ – другой общий делитель чисел $a_1, a_2, \ldots, a_k$, то d делится на d_1.

Например, НОД(12345, 24690) = 12345, НОД(12345, 54321) = 3, НОД(12345, 12541) = 1.

Ненулевые целые числа a и b называются *ассоциированными* (обозначается $a \sim b$), если a делится на b и b делится на a.

Для любых целых чисел $a_1, a_2, \ldots, a_k$ существует наибольший общий делитель d и его можно представить в виде *линейной комбинации* этих чисел:

$$d = c_1 a_1 + c_2 a_2 + \cdots + c_k a_k,\ c_i \in Z\ (Z \text{ – множество целых чисел}).$$

Например, НОД чисел 91, 105, 154 равен 7. В качестве линейного представления можно взять

$$7 = 7 \cdot 91 + (-6) \cdot 105 + 0 \cdot 154,$$

либо

$$7 = 4 \cdot 91 + 1 \cdot 105 - 3 \cdot 154.$$

Целые числа $a_1, a_2, \ldots, a_k$ называются *взаимно простыми в совокупности*, если НОД$(a_1, a_2, \ldots, a_k)$=1. Целые числа a и b называются *взаимно простыми*, если НОД(a,b)=1.

Целые числа $a_1, a_2, \ldots, a_k$ называются *попарно взаимно простыми*, если НОД(a_i, a_j)=1 для всех $1 \leq i \neq j \leq k$.

Алгоритмы вычисления наибольшего общего делителя.

Для вычисления наибольшего общего делителя двух целых чисел применяется способ повторного деления с остатком, называемый *алгоритмом Евклида*.

1. Алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$.

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, i \leftarrow 1$.
2. Найти остаток r_{i+1}от деления r_{i-1}наr_i.
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$. В противном случае положить $i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d.

Бинарный алгоритм Евклида является более быстрым при реализации на компьютере, поскольку использует двоичное представление чисел a и b. Бинарный алгоритм Евклида основан на следующих свойствах наибольшего общего делителя (считаем, что $0 < b \leq a$):

1) если оба числа a и b четные, то $\text{НОД}(a, b) = 2 \cdot \text{НОД}(\frac{a}{2}, \frac{b}{2})$;
2) если число a – нечетное, число b – четное, то $\text{НОД}(a, b) = \text{НОД}(a, \frac{b}{2})$;
3) если оба числа a и b нечетные, $a > b$, то $\text{НОД}(a, b) = \text{НОД}(a - b, b)$;
4) если $a = b$, то $\text{НОД}(a, b) = a$.

2. Бинарный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$.

1. Положить $g \leftarrow 1$.
2. Пока оба числа a и b четные, выполнять $a \leftarrow \frac{a}{2}$, $b \leftarrow \frac{b}{2}$, $g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b.
3. Положить $u \leftarrow a, v \leftarrow b$.
4. Пока $u \neq 0$ выполнять следующие действия:

4.1.Пока uчетное, полагать $u \leftarrow \frac{u}{2}$.

4.2.Пока vчетное, полагать $v \leftarrow \frac{v}{2}$.

4.3.При $u \geq v$ положить $u \leftarrow u - v$. В противном случае положить $v \leftarrow v - u$.

5. Положить $d \leftarrow gv$.
6. Результат: d

3. Расширенный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$; такие целые числа x, y, что $ax + by = d$.

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, x_0 \leftarrow 1, x_1 \leftarrow 0, y_0 \leftarrow 0, y_1 \leftarrow 1, i \leftarrow 1$.
2. Разделить с остатком r_{i-1} на r_i: $r_{i-1} = q_i r_i + r_{i+1}$.
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$, $x \leftarrow x_i$, $y \leftarrow y_i$. В противном случае положить $x_{i+1} \leftarrow x_{i-1} - q_i x_i, y_{i+1} \leftarrow y_{i-1} - q_i y_i, i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d, x, y.

4. Расширенный бинарный алгоритм Евклида.

Вход. Целые числа a, b; $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$.

1. Положить $g \leftarrow 1$.
2. Пока числа a и b четные, выполнять $a \leftarrow \frac{a}{2}$, $b \leftarrow \frac{b}{2}$, $g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b.
3. Положить $u \leftarrow a, v \leftarrow b, A \leftarrow 1, B \leftarrow 0, C \leftarrow 0, D \leftarrow 1$.
4. Пока $u \neq 0$ выполнять следующие действия:

   4.1.Пока u четное:

   4.1.1. Положить $u \leftarrow \frac{u}{2}$.

   4.1.2. Если оба числа A и B четные, то положить $A \leftarrow \frac{A}{2}, B \leftarrow \frac{B}{2}$. В противном случае положить $A \leftarrow \frac{A+b}{2}$, $B \leftarrow \frac{B-a}{2}$.

   4.2.Пока v четное:

   4.2.1. Положить $v \leftarrow \frac{v}{2}$.

4.2.2. Если оба числа C и D четные, то положить $C \leftarrow \frac{C}{2}$, $D \leftarrow \frac{D}{2}$. В противном случае положить $C \leftarrow \frac{C+b}{2}, D \leftarrow \frac{D-a}{2}$.

4.3. При $u \geq v$ положить $u \leftarrow u - v, A \leftarrow A - C, B \leftarrow B - D$. В противном случае положить $v \leftarrow v - u, C \leftarrow C - A, D \leftarrow D - B$.

5. Положить $d \leftarrow gv, x \leftarrow C, y \leftarrow D$.
6. Результат: d, x, y.

Задания к лабораторной работе

Реализовать все рассмотренные алгоритмы программно.

ЛАБОРАТОРНАЯ РАБОТА №5

Вероятностные алгоритмы проверки чисел на простоту

Пусть a – целое число. Числа $\pm 1, \pm a$ называются *тривиальными делителями* числа a.

Целое число $p \in Z/\{0\}$ называется *простым*, если оно не является делителем единицы и не имеет других делителей, кроме тривиальных. В противном случае число $p \in Z/\{-1, 0, 1\}$ называется *составным*.

Например, числа $\pm 2, \pm 3, \pm 5, \pm 7, \pm 11, \pm 13, \pm 17, \pm 19, \pm 23, \pm 29$ являются простыми.

Пусть $m \in N, m > 1$. Целые числа a и b называются сравнимыми по модулю m (обозначается $a \equiv b \ (mod\ m)$) если разность $a - b$ делится на m. Также эта процедура называется нахождением остатка от целочисленного деления a на b.

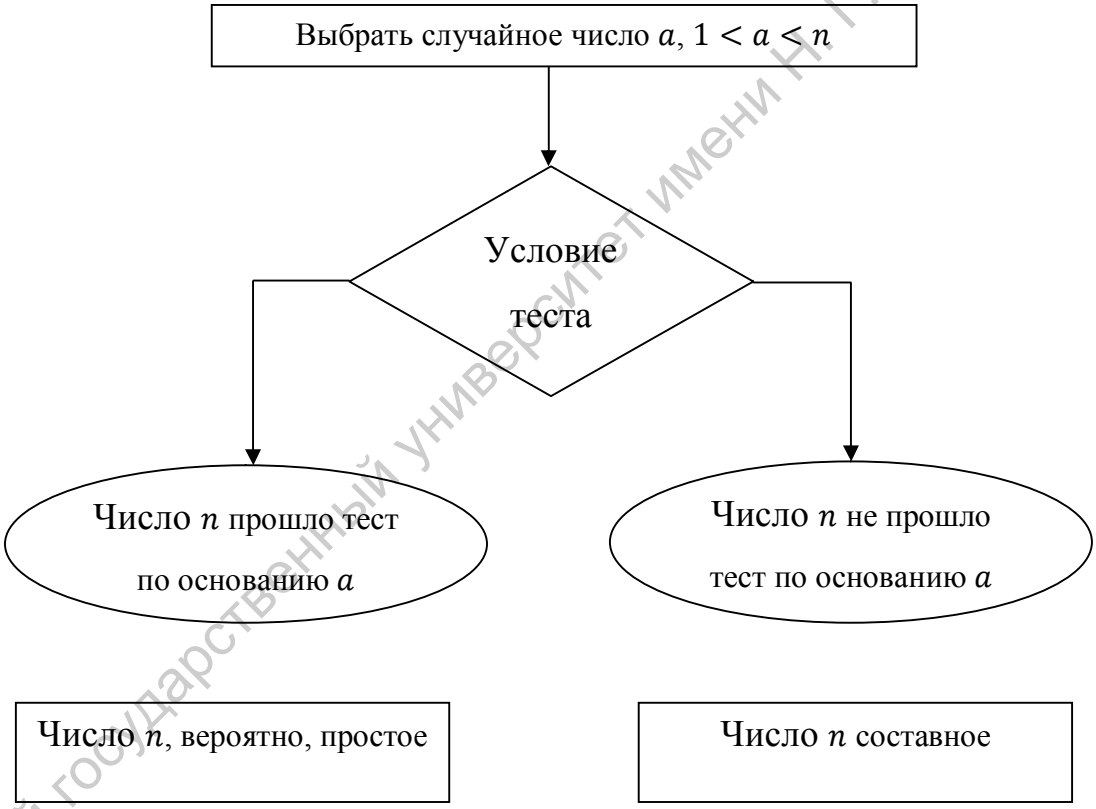
Проверка чисел на простоту является составной частью алгоритмов генерации простых чисел, применяемых в криптографии с открытым ключом. Алгоритмы проверки на простоту можно разделить на вероятностные и детерминированные.

Детерминированный алгоритм всегда действует по одной и той же схеме и гарантированно решает поставленную задачу (или не дает никакого ответа). *Вероятностный* алгоритм использует генератор случайных чисел и дает не гарантированно точный ответ. Вероятностные алгоритмы в общем случае не менее эффективны, чем детерминированные (если используемый генератор случайных чисел всегда дает набор одних и тех же чисел, зависящих от входных данных, то вероятностный алгоритм становится детерминированным).

Для проверки на простоту числа n вероятностным алгоритмом выбирают случайной число a $(1 < a < n)$ и проверяют условия алгоритма. Если число n не проходит тест по основанию a, то алгоритм выдает результат «Число n составное», и число n действительно является составным.

Если же n проходит тест по основанию a, ничего нельзя сказать о том, действительно ли число n является простым. Последовательно проведя ряд проверок таким тестом для разных a и получив для каждого из них ответ «Число n, вероятно, простое», можно утверждать, что число n является простым с вероятностью, близкой к 1. После t независимых выполнений теста вероятность того, что составное число n будет t раз объявлено простым (вероятность ошибки), не превосходит $\frac{1}{2^t}$.

Схема вероятностного алгоритма проверки числа на простоту



Тест Ферма основан на малой теореме Ферма: для простого числа p и произвольного числа a, $1 \le a \le p-1$, выполняется сравнение

$$a^{p-1} \equiv 1 \ (mod\ p).$$

Следовательно, если для нечетного n существует такое целое a, что $1 \le a < n$, НОД$(a,n) = 1$ и $a^{n-1} \neq 1 \ (mod\ n)$, то число n составное. Отсюда получаем следующий вероятностный алгоритм проверки числа на простоту.

1. Алгоритм, реализующий тест Ферма.

Вход. Нечетное целое число $n \geq 5$.

Выход. «Число n, вероятно, простое» или «Число n составное».

1. Выбрать случайное целое число a, $2 \leq a \leq n-2$.
2. Вычислить $r \leftarrow a^{n-1} \ (mod\ n)$.
3. При $r = 1$ результат: «Число n, вероятно, простое». В противном случае результат: «Число n составное».

На шаге 1 мы не рассматривали числа $a = 1$ и $a = n-1$, поскольку $1^{n-1} \equiv 1\ (mod\ n)$ для любого целого n и $(n-1)^{n-1} \equiv (-1)^{n-1} \equiv 1\ (mod\ n)$ для любого нечетного n.

Тест Соловэя-Штрассена. Основан на критерии Эйлера: нечетное число n является простым тогда и только тогда, когда для любого целого числа a, $1 \leq a \leq n-1$, взаимно простого с n, выполняется сравнение:

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right)\ (mod\ n),$$

где $\left(\frac{a}{n}\right)$ – *символ Якоби*.

Пусть $m, n \in Z$, где $n = p_1 p_2 \dots p_r$ и числа $p_i \neq 2$ простые (не обязательно различные). *Символ Якоби* $\left(\frac{m}{n}\right)$ определяется равенством

$$\left(\frac{m}{n}\right) = \left(\frac{m}{p_1}\right)\left(\frac{m}{p_2}\right)\dots\left(\frac{m}{p_r}\right).$$

2. Алгоритм вычисления символа Якоби.

Вход. Нечетное целое число $n \geq 3$, целое число a, $0 \leq a < n$.

Выход. Символ Якоби $\left(\frac{a}{n}\right)$.

1. Положить $g \leftarrow 1$.

2. При $a = 0$ результат: 0.
3. При $a = 1$ результат: g.
4. Представить a в виде $a = 2^k a_1$, где число a_1 нечетное.
5. При четном k положить $s \leftarrow 1$, при нечетном k положить $s \leftarrow 1$, если $n \equiv \pm 1 \ (mod\ 8)$; положить $s \leftarrow -1$, если $n \equiv \pm 3 \ (mod\ 8)$.
6. При $a_1 = 1$ результат: $g \cdot s$.
7. Если $n \equiv 3 \ (mod\ 4)$ и $a_1 \equiv 3 \ (mod\ 4)$, то $s \leftarrow -s$.
8. Положить $a \leftarrow n \ (mod\ a_1)$, $n \leftarrow a_1, g \leftarrow g \cdot s$ и вернуться на шаг 2.

3. Алгоритм, реализующий тест Соловэя-Штрассена.

Вход. Нечетное целое число $n \geq 5$.

Выход. «Число n, вероятно, простое» или «Число n составное».

1. Выбрать случайное целое число a, $2 \leq a < n - 2$.
2. Вычислить $r \leftarrow a^{\frac{n-1}{2}} \ (mod\ n)$.
3. При $r \neq 1$ и $r \neq n - 1$ результат: «Число n составное».
4. Вычислить *символ Якоби* $s \leftarrow \left(\frac{a}{n}\right)$.
5. При $r \equiv s \ (mod\ n)$ результат: «Число n составное». В противном случае результат: «Число n, вероятно, простое».

На сегодняшний день для проверки чисел на простоту чаще всего используется тест Миллера-Рабина, основанный на следующем наблюдении. Пусть число n нечетное и $n - 1 = 2^s r$, где r – нечетное. Если n простое, то для любого $a \geq 2$, взаимно простого с n, выполняется условие $a^{p-1} \equiv 1 \ (mod\ p)$.

4. Алгоритм, реализующий тест Миллера-Рабина.

Вход. Нечетное целое число $n \geq 5$.

Выход. «Число n, вероятно, простое» или «Число n составное».

1. Представить $n - 1$ в виде $n - 1 = 2^s r$, где число r нечетное.
2. Выбрать случайное целое число a, $2 \leq a < n - 2$.

3. Вычислить $y \leftarrow a^r (mod\ n)$.
4. При $y \neq 1$ и $y \neq n-1$ выполнить следующие действия.

   4.1. Положить $j \leftarrow 1$.

   4.2. Если $j \leq s-1$ и $y \neq n-1$, то

      4.2.1. Положить $y \leftarrow y^2\ (mod\ n)$.

      4.2.2. При $y = 1$ результат: «Число n составное».

      4.2.3. Положить $j \leftarrow j+1$.

   4.3. При $y \neq n-1$ результат: «Число n составное».
5. Результат: «Число n, вероятно, простое».

Задания к лабораторной работе

Реализовать все рассмотренные алгоритмы программно.

ЛАБОРАТОРНАЯ РАБОТА №6

Разложение чисел на множители

Задача разложения на множители – одна из первых задач, использованных для построения криптосистем с открытым ключом.

Задача разложения составного числа на множители формулируется следующим образом: для данного положительного целого числа n найти его каноническое разложение $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$, где p_i – попарно различные простые числа, $\alpha_i \geq 1$.

На практике не обязательно находить каноническое разложение числа n. Достаточно найти его разложение на два *нетривиальных сомножителя*: $n = pq, 1 \leq p \leq q < n$. Далее будем понимать задачу разложения именно в этом смысле.

p–Метод Полларда. Пусть n – нечетное составное число, $S = \{0, 1, \dots, n-1\}$ и f: $S \to S$ – случайное отображение, обладающее сжимающими свойствами, например $f(x) \equiv x^2 + 1\ (mod\ n)$. Основная идея метода состоит в следующем. Выбираем случайный элемент $x_0 \in S$ и строим последовательность $x_0, x_1, x_{2,\dots}$, определяемую рекуррентным соотношением

$$x_{i+1} = f(x_i),$$

где $i \geq 0$, до тех пор, пока не найдем такие числа i, j, что $i < j$ и $x_i = x_j$. Поскольку множество S конечно, такие индексы i, j существуют (последовательность «зацикливается»). Последовательность $\{x_i\}$ будет состоять из «хвоста» $x_0, x_1, \dots, x_{i-1}$ длины $O\left(\sqrt{\frac{\pi n}{8}}\right)$ и цикла $x_i = x_j, x_{i+1}, \dots, x_{j-1}$ той же длины.

Алгоритм, реализующий p-метод Полларда.

Вход. Число n, начальное значение c, функция f, обладающая сжимающими свойствами.

Выход. Нетривиальный делитель числа n.

1. Положить $a \leftarrow c, b \leftarrow c$.
2. Вычислить $a \leftarrow f(a)(mod\ n), b \leftarrow f(b)\ (mod\ n)$
3. Найти $d \leftarrow$ НОД$(a - b, n)$.
4. Если $1 < d < n$, то положить $p \leftarrow d$ и результат: p. При $d = n$ результат: «Делитель не найден»; при $d = 1$ вернуться на шаг 2.

Пример. Найти р-методом Полларда нетривиальный делитель числа $n = 1359331$. Положим $c = 1$ и $f(x) = x^2 + 5\ (mod\ n)$. Работа алгоритма иллюстрируется следующей таблицей:

i	a	b	$d = \text{НОД}(a - b, n)$
	1	1	
2	6	41	1
2	41	123939	1
3	1686	391594	1
4	123939	438157	1
5	435426	582738	1
6	391594	1144026	1
7	1090062	885749	1181

Таким образом, 1181 является нетривиальным делителем числа 1359331.

Метод квадратов. (Теорема Ферма о разложении) Для любого положительного нечетного числа n существует взаимно однозначное соответствие между множеством делителей числа n, не меньших, чем $\sqrt{n}$, и множеством пар $\{s, t\}$ таких неотрицательных целых чисел, что $n = s^2 - t^2$.

Пример. У числа 15 два делителя, не меньших, чем $\sqrt{15}$, – это числа 5 и 15. Тогда получаем два представления:

1. $15 = pq = 3 \cdot 5$, откуда $s = 4, t = 1$ и $15 = 4^2 - 1^2$;
2. $15 = pq = 1 \cdot 15$, откуда $s = 8, t = 7$ и $15 = 8^2 - 7^2$.

Задания к лабораторной работе

1. Реализовать рассмотренный алгоритм программно.
2. Разложить на множители данное преподавателем число.

ЛАБОРАТОРНАЯ РАБОТА №7

Дискретное логарифмирование в конечном поле

Задача дискретного логарифмирования, как и задача разложения на множители, применяется во многих алгоритмах криптографии с открытым ключом. Предложенная в 1976 году У. Диффи и М. Хеллманом для установления сеансового ключа, эта задача послужила основой для создания протоколов шифрования и цифровой подписи, доказательств с нулевым разглашением и других криптографических протоколов.

Пусть над некоторым множеством Ω произвольной природы определены операции сложения «+» и умножения «·». Множество Ω называется *кольцом*, если выполняются следующие условия:

1. Сложение коммутативно: $a + b = b + a$ для любых $a, b \in \Omega$;
2. Сложение ассоциативно: $(a + b) + c = a + (b + c)$ для любых $a, b, c \in \Omega$;
3. Существует нулевой элемент $0 \in \Omega$ такой, что $a + 0 = a$ для любого $a \in \Omega$;
4. Для каждого элемента $a \in \Omega$ существует противоположный элемент $-a \in \Omega$, такой, что $(-a) + a = 0$;
5. Умножение дистрибутивно относительно сложения:

$$a \cdot (b + c) = a \cdot b + a \cdot c, (a + b) \cdot c = a \cdot c + b \cdot c,$$

для любых $a, b, c \in \Omega$.

Если в кольце Ω умножение коммутативно: $a \cdot b = b \cdot a$ для любых $a, b \in \Omega$, то кольцо называется *коммутативным*.

Если в кольце Ω умножение ассоциативно: $(a \cdot b) \cdot \mathrm{c} = a \cdot (b \cdot c)$ для любых $a, b, c \in \Omega$, то кольцо называется *ассоциативным*.

Если в кольце Ω существует единичный элемент e такой, что $a \cdot e = e \cdot a = a$ для любого $a \in \Omega$, то кольцо называется кольцом с единицей.

Если в ассоциативном, коммутативном кольце Ω с единицей для каждого ненулевого элемента a существует обратный элемент $a^{-1} \in \Omega$ такой, что $a^{-1} \cdot a = a \cdot a^{-1} = e$, то кольцо называется *полем*.

Пусть $m \in N, m > 1$. Целые числа a и b называются *сравнимыми по модулю* m (обозначается $a \equiv b \ (mod\ m)$), если разность $a - b$ делится на m. Некоторые свойства отношения сравнимости:

1. *Рефлексивность:* $a \equiv a \ (mod\ m)$.
2. *Симметричность*: если $a \equiv b \ (mod\ m)$, то $b \equiv a \ (mod\ m)$.
3. *Транзитивность:* если $a \equiv b \ (mod\ m)$ и $b \equiv c \ (mod\ m)$, то $a \equiv c \ (mod\ m)$.

Отношение, обладающее свойством рефлексивности, симметричности и транзитивности, называется *отношением эквивалентности*. Отношение сравнимости является отношением эквивалентности на множестве Z целых чисел.

Отношение эквивалентности разбивает множество, на котором оно определено, на *классы эквивалентности.* Любые два класса эквивалентности либо не пересекаются, либо совпадают.

Классы эквивалентности, определяемые отношением сравнимости, называются *классами вычетов по модулю* m. Класс вычетов, содержащий число a, обозначается $a \ (mod\ m)$ или $\bar{a}$ и представляет собой множество чисел вида $a + km$, где $k \in Z$; число a называется представителем этого класса вычетов.

Множество классов вычетов по модулю m обозначается Z/mZ, состоит ровно из m элементов и относительно операций сложения и умножения является *кольцом классов вычетов* по модулю m.

Пример. Если $m = 2$, то Z/2Z= $\{0 \ (mod\ 2), 1 \ (mod\ 2)\}$, где $0 \ (mod\ 2) = 2Z$ – множество всех четных чисел, $1 \ (mod\ 2) = 2Z + 1$ – множество всех нечетных чисел.

Обозначим $F_p = Z/pZ$, p – простое целое число и назовем конечным полем из p элементов. Задача дискретного логарифмирования в конечном поле F_p

формулируется так: для данных целых чисел a и b, $a > 1, b > p$, найти логарифм – такое целое число x, что $a^x \equiv b\ (mod\ p)$ (если такое число существует). По аналогии с вещественными числами используется обозначение $x = \log_a b$.

Безопасность соответствующих криптосистем основана на том, что, зная числа a, x, p вычислить $a^x\ (mod\ p)$ легко, а решит задачу дискретного логарифмирования трудно. Рассмотрим р–Метод Полларда, который можно применить и для задач дискретного логарифмирования. При этом случайное отображение f должно обладать не только сжимающими свойствами, но и вычислимостью логарифма (логарифм числа $f(c)$ можно выразить через неизвестный логарифм x и $\log_a f(c)$). Для дискретного логарифмирования в качестве случайного отображения f чаще всего используются ветвящиеся отображения, например:

$$f(c) = \begin{cases} ac, \text{при } c < \frac{p}{2} \\ bc, \text{при } c > \frac{p}{2} \end{cases}.$$

При $c < \frac{p}{2}$ имеем $\log_a f(c) = \log_a c + 1$, при $c > \frac{p}{2}$ – $\log_a f(c) = \log_a c + x$.

Алгоритм, реализующий р–Метод Полларда для задач дискретного логарифмирования.

Вход. Простое число p, число a порядка r по модулю p, целое число $b, 1 < b < p$; отображение f, обладающее сжимающими свойствами и сохраняющее вычислимость логарифма.

Выход. Показатель x, для которого $a^x \equiv b\ (mod\ p)$, если такой показатель существует.

1. Выбрать произвольные целые числа u, v и положить $c \leftarrow a^u b^v\ (mod\ p)$, $d \leftarrow c$.
2. Выполнять $c \leftarrow f(c)(mod\ p), d \leftarrow f\big(f(d)\big)(mod\ p)$, вычисляя при этом логарифмы для c и d как линейные функции от x по модулю r, до получения равенства $c \equiv d\ (mod\ p)$.
3. Приравняв логарифмы для c и d, вычислить логарифм x решением сравнения по модулю r. Результат: x или "Решений нет".

Пример. Решим задачу дискретного логарифмирования $10^x \equiv 64\ (mod\ 107)$, используя p–Метод Полларда. Порядок числа 10 по модулю 107 равен 53.

Выберем отображение $f(c) \equiv 10c\ (mod\ 107)$ при $c < 53$, $f(c) \equiv 64c\ (mod\ 107)$ при $c \geq 53$. Пусть $u = 2, v = 2$. Результаты вычислений запишем в таблицу:

Номер шага	c	$\log_a c$	d	$\log_a d$
0	4	2+2 x	4	2+2 x
1	40	3+2 x	76	4+2 x
2	79	4+2 x	56	5+3 x
3	27	4+3 x	75	5+5 x
4	56	5+3 x	3	5+7 x
5	53	5+4 x	86	7+7 x
6	75	5+5 x	42	8+8 x
7	92	5+6 x	23	9+9 x
8	3	5+7 x	53	11+9 x
9	30	6+7 x	92	11+11 x
10	86	7+7 x	30	12+12 x
11	47	7+8 x	47	13+13 x

Приравниваем логарифмы, полученные на 11-м шаге: 7+8 x ≡13+13 x $(mod\ 53)$. Решая сравнение первой степени, получаем: $x = 20\ (mod\ 53)$.

Проверка: $10^{20} \equiv 64\ (mod\ 107)$.

Задания к лабораторной работе

1. Реализовать алгоритм программно.
2. Получить у преподавателя задание, содержащее числа p, a, b и вычислить логарифм.

ЛАБОРАТОРНАЯ РАБОТА №8

Целочисленная арифметика многократной точности

В данной работе рассмотрим алгоритмы для выполнения арифметических операций с большими целыми числами. Будем считать, что число записано в b-ичной системе счисления, b – натуральное число, $b \geq 2$. Натуральное -разрядное число будем записывать в виде

$$u = u_1 u_2 \dots u_n.$$

При работе с большими целыми числами знак такого числа удобно хранить в отдельной переменной. Например, при умножении двух чисел, знак произведения вычисляется отдельно. Квадратные скобки обозначают, что берется целая часть числа.

Алгоритм 1 (сложение неотрицательных целых чисел).

Вход. Два неотрицательных числа $u = u_1 u_2 \dots u_n$ и $v = v_1 v_2 \dots v_n$; разрядность чисел n; основание системы счисления b.

Выход. Сумма $w = w_0 w_1 \dots w_n$, где w_0 – цифра переноса – всегда равная 0 либо 1.

1. Присвоить $j := n$, $k := 0$ (j идет по разрядам, k следит за переносом).
2. Присвоить $w_j = (u_j + v_j + k) \ (mod\ b)$, где w_j – наименьший неотрицательный вычет в данном классе вычетов; $k = \left[\frac{u_j + v_j + k}{b}\right]$.
3. Присвоить $j := j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то присвоить $w_0 := k$ и результат: w.

Алгоритм 2 (вычитание неотрицательных целых чисел).

Вход. Два неотрицательных числа $u = u_1 u_2 \dots u_n$ и $v = v_1 v_2 \dots v_n$, $u > v$; разрядность чисел n; основание системы счисления b.

Выход. Разность $w = w_1 w_2 \dots w_n = u - v$.

1. Присвоить $j := n$, $k := 0$ (k – заем из старшего разряда).

2. Присвоить $w_j = (u_j - v_j + k)\ (mod\ b)$, где w_j – наименьший неотрицательный вычет в данном классе вычетов; $k = \left[\frac{u_j - v_j + k}{b}\right]$.
3. Присвоить $j \coloneqq j - 1$. Если $j > 0$, то возвращаемся на шаг 2; если $j = 0$, то результат: w.

Алгоритм 3 (умножение неотрицательных целых чисел столбиком).

Вход. Числа $u = u_1u_2 \dots u_n$, $v = v_1v_2 \dots v_m$; основание системы счисления b.

Выход. Произведение $w = uv = w_1w_2 \dots w_{m+n}$.

1. Выполнить присвоения: $w_{m+1} \coloneqq 0, w_{m+2} \coloneqq 0, \dots, w_{m+n} \coloneqq 0, j \coloneqq m$ (j перемещается по номерам разрядов числа v от младших к старшим).
2. Если $v_j = 0$, то присвоить $w_j \coloneqq 0$ и перейти на шаг 6.
3. Присвоить $i \coloneqq n, k \coloneqq 0$ (Значение i идет по номерам разрядов числа u, k отвечает за перенос).
4. Присвоить $t \coloneqq u_i \cdot v_j + w_{i+j} + k,\ w_{i+j} \coloneqq t\ (mod\ b),\ k \coloneqq \frac{t}{b}$, где w_{i+j} – наименьший неотрицательный вычет в данном классе вычетов.
5. Присвоить $i \coloneqq i - 1$. Если $i > 0$, то возвращаемся на шаг 4, иначе присвоить $w_j \coloneqq k$.
6. Присвоить $j \coloneqq j - 1$. Если $j > 0$, то вернуться на шаг 2. Если $j = 0$, то результат: w.

Алгоритм 4 (быстрый столбик).

Вход. Числа $u = u_1u_2 \dots u_n$, $v = v_1v_2 \dots v_m$; основание системы счисления b.

Выход. Произведение $w = uv = w_1w_2 \dots w_{m+n}$.

1. Присвоить $t \coloneqq 0$.
2. Для s от 0 до $m + n - 1$ с шагом 1 выполнить шаги 3 и 4.
3. Для i от 0 до s с шагом 1 выполнить присвоение $t \coloneqq t + u_{n-i} \cdot v_{m-s+i}$.
4. Присвоить $w_{m+n-s} \coloneqq t\ (mod\ b),\ t \coloneqq \frac{t}{b}$, где w_{m+n-s} – наименьший неотрицательный вычет по модулю b. Результат: w.

Алгоритм 5 (деление многоразрядных целых чисел).

Вход. Числа $u = u_n \dots u_1 u_0$, $v = v_t \dots v_1 v_0$, $n \geq t \geq 1$, $v_t \neq 0$, разрядность чисел соответственно n и t.

Выход. Частное $q = q_{n-t} \dots q_0$, остаток $r = r_t \dots r_0$.

1. Для j от 0 до $n - t$ присвоить $q_j := 0$.
2. Пока $u \geq vb^{n-t}$, выполнять: $q_{n-t} := q_{n-t} + 1$, $u := u - vb^{n-t}$.
3. Для $i = n, n-1, \dots, t+1$ выполнять пункты 3.1 – 3.4:

   3.1 если $u_i \geq v_t$, то присвоить $q_{i-t-1} := b - 1$, иначе присвоить $q_{i-t-1} := \frac{u_i b + u_{i-1}}{v_t}$.

   3.2 пока $q_{i-t-1}(v_t b + v_{t-1}) > u_i b^2 + u_{i-1} b + u_{i-2}$ выполнять $q_{i-t-1} := q_{i-t-1} - 1$.

   3.3 присвоить $u := u - q_{i-t-1} b^{i-t-1} v$.

   3.4 если $u < 0$, то присвоить $u := u + vb^{i-t-1}$, $q_{i-t-1} := q_{i-t-1} - 1$.
4. $r := u$. Результат: q и r.

Задания к лабораторной работе

Реализовать рассмотренные алгоритмы программно.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Кнут Д. Искусство программирования для ЭВМ. Т.2. Получисленные алгоритмы / Д. Кнут. — М.: Мир, 1977. — 724 с.
2. Молдовян А.А. Криптография /А.А. Молдовян Н.А. Молдовян, Б.Я. Советов. — СПб.: «Лань», 2000. — 224 с.
3. Ахо А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. — М.: Мир, 1979. — 535 с.
4. Адигеев М.Г. Введение в криптографию. Ч.1. Основные понятия, задачи и методы криптографии / М.Г. Адигеев. – Ростов – на – Дону, 2002 г. – 35 с.
5. Асосков А.В. Поточные шифры / А.В. Асосков, М.А. Иванов, А.А. Мирский, А.В. Рузин, А.В. Сланин, А.Н. Тютвин. – М.: «КУДИЦ-ОБРАЗ», 2003 г. – 334 с.
6. Аграновский А.В. Практическая криптография: алгоритмы и их пограммирование / А.В. Аграновский, Р.А. Хади. – М.:СОЛОН-Пресс, 2009 г. – 256 с.
7. Салий В.Н. Криптографические методы и средства защиты информации / В.Н. Салий. – 40 с.
8. Маховенко Е.Б. Теоретико-числовые методы в криптографии / Е.Б. Маховенко. – М.: Гелиос АРВ, 2006 г., 320 с., ил.
9. Фомичев В.М. Дискретная математика и криптология / В.М. Фомичев. – М.: ДИАЛОГ МИФИ, 2003 г., 397 с.

Учебно-методическое издание

Бубнов Сергей Алексеевич

Лабораторный практикум по основам криптографии

Учебно-методическое пособие
для студентов по профилю подготовки 080801.65 «Прикладная информатика (в экономике)»