Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет инфокоммуникаций

Кафедра защиты информации

А. М. Тимофеев

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия для специальности 1-98 01 02 «Защита информации в телекоммуникациях»

Минск БГУИР 2018

УДК 004.056.55(076)
ББК 32.972.5я73
Т41

Рецензенты:

кафедра телекоммуникационных систем учреждения образования «Белорусская государственная академия связи» (протокол №1 от 30.08.2017);

доцент кафедры информационно-измерительной техники и технологии Белорусского национального технического университета, кандидат технических наук, доцент А. К. Тявловский;

доцент кафедры инфокоммуникационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент А. Е. Лагутин

Тимофеев, А. М.

Т41 Криптографическая защита информации : пособие / А. М. Тимофеев. – Минск : БГУИР, 2018. – 44 с. : ил.

ISBN 978-985-543-399-7.

Пособие содержит пять тем практических занятий и направлено на изучение используемых в криптографии арифметических операций и операторов в системе наименьших вычетов, а также простых чисел и их применения в асимметрично-ключевой криптографии. Композиционно каждое практическое занятие включает краткие теоретические сведения, практическое задание, содержание отчета и перечень контрольных вопросов.

Предназначено для студентов специальности 1-98 01 02 «Защита информации в телекоммуникациях» дневной формы обучения, может быть полезно студентам и магистрантам инфокоммуникационных специальностей всех форм обучения, а также специалистам, работающим в области проектирования и создания систем защиты информации.

УДК 004.056.55(076)
ББК 32.972.5я73

ISBN 978-985-543-399-7

СОДЕРЖАНИЕ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1
ИЗУЧЕНИЕ ОСНОВНЫХ ИСПОЛЬЗУЕМЫХ В КРИПТОГРАФИИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Цель: изучение основных используемых в криптографии арифметических операций.

1.1 Краткие теоретические сведения

В криптографии в арифметике целых используется множество целых чисел Z и следующие основные операции [1–7]:

– бинарные операции (сложение, вычитание и умножение);

– операция деления;

– операции нахождения наибольшего общего делителя чисел (НОД).

Множество целых чисел Z содержит все числа (без дробей) от минус бесконечности до плюс бесконечности и может быть записано как

$$Z = \{-\infty, \ldots, -2,\ -1,\ 0,\ 1,\ 2, \ldots, +\infty\}. \qquad (1)$$

Бинарные операции имеют два входа и один выход.

Деление числа a на n в арифметике целых чисел можно представить уравнением деления:

$$a = q \cdot n + r, \qquad (2)$$

где a – делимое;

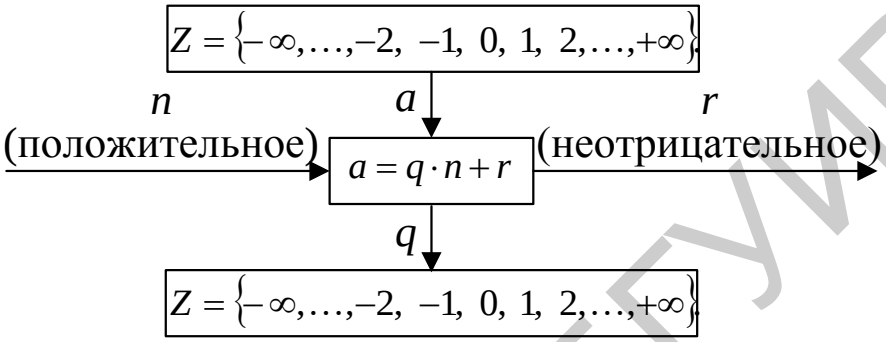
q – частное;

n – делитель;

r – остаток.

При использовании уравнения деления (2) в криптографии есть два ограничения: делитель должен быть положительным целым числом ($n > 0$), а остаток – неотрицательным целым числом ($r \geq 0$).

На рисунке 1 показан алгоритм деления целых чисел.

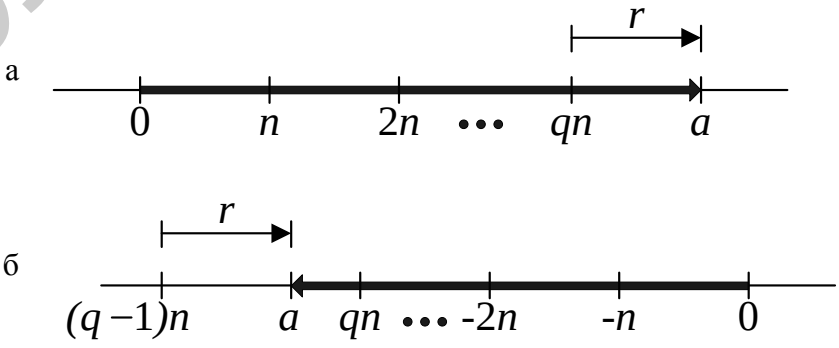


a и *b* – входы;

c – выход

Рисунок 1 – Алгоритм деления целых чисел

При расчетах указанные ограничения можно выполнить путем уменьшения значение *q* на 1 и добавления значение *n* к *r*, чтобы *r* стало положительным. Уравнения деления с ограничениями на *n* и *r* можно изобразить с помощью двух графов, приведенных на рисунке 2.



а – число *a* является положительным;

б – число *a* является отрицательным

Рисунок 2 – Граф алгоритма деления

Если число a не равно нулю и делится на n без остатка ($r = 0$), то говорят, что «a делится на n» (или «n – делитель a»), что записывается как $a|n$. Если остаток не является нулевым ($r \neq 0$), то говорят, что «n не делит a», что записывается как $a \nmid n$.

Приведем основные свойства теории делимости:

– если $a|1$, то $a = \pm 1$;

– если $a|b$ и $b|a$, то $a = \pm b$;

– если $a|b$ и $b|c$, то $a|c$;

– если $a|b$ и $a|c$, то $a|(m \cdot b + n \cdot c)$, где m и n – произвольные целые числа.

В криптографии часто требуется найти НОД двух положительных целых чисел.

НОД двух положительных целых чисел – это наибольшее целое число, которое делит оба целых числа.

Нахождение НОД двух положительных целых чисел путем составления списка всех общих делителей непригодно для достаточно больших чисел, поэтому чаще используют специальные алгоритмы: алгоритм Евклида и расширенный алгоритм Евклида.

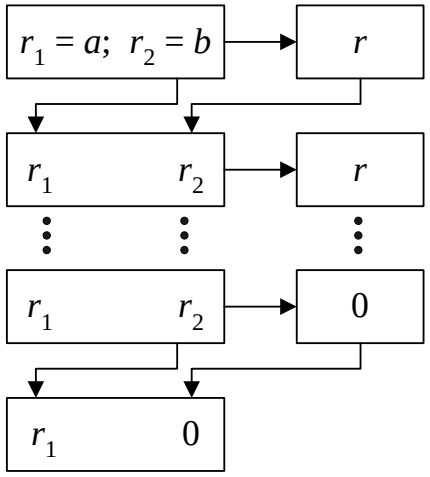
Алгоритм Евклида основан на следующих двух фактах: НОД (a, 0) = a и НОД (a, b) = НОД (b, r), где r – остаток от деления a на b. Первый факт говорит, что если второе целое число равно нулю, то НОД равен первому числу. Второй факт позволяет изменять значение a на b, пока b не станет равно нулю.

На рисунке 3 показан принцип реализации алгоритма Евклида.

Расширенный алгоритм Евклида применяется, когда известны два числа a и b и требуется найти другие два целых числа s и t, при которых выполняется равенство

$$s \cdot a + t \cdot b = \text{НОД}(a, b). \qquad (3)$$

а



б

```
r1 ← a;  r2 ← b (инициализация)

while (r2 > 0)
{
q ← r1 / r2;

r ← r1 – q · r2;
r1 ← r2; r2← r;
}
НОД (a, b) = r1
```

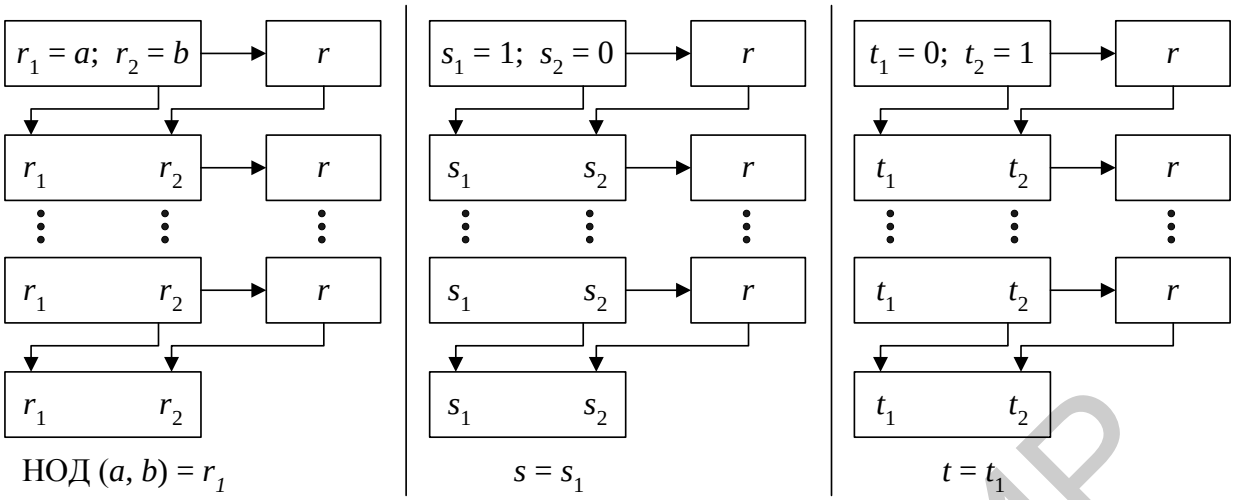
Рисунок 3 – Блок-схема (а) и программная реализация (б) алгоритма Евклида

Расширенный алгоритм Евклида позволяет вычислить НОД (a, b) и определить значения s и t.

Расширенный алгоритм Евклида использует те же самые шаги, что и простой алгоритм Евклида (см. рисунок 3), однако на каждой итерации применяются три группы вычислений вместо одной и используются три набора переменных: r, s и t. На каждом шаге переменные r_1, r_2 и r используются так же, как в алгоритме Евклида. Переменным r_1 и r_2 присваиваются начальные значения a и b соответственно. Переменным s_1 и s_2 присваиваются начальные значения 1 и 0 соответственно. Переменным t_1 и t_2 присваиваются начальные значения 0 и 1 соответственно. Вычисления r, s и t одинаковы, но с одним отличием. Хотя r является остатком от деления r_1 на r_2, такого соответствия в других двух группах вычислений нет. Есть только одно частное q, которое вычисляется как r_1/r_2 и используется для других двух вычислений.

На рисунке 4 показан принцип реализации расширенного алгоритма Евклида.

а



б

```
r1 ← a; r2 ← b;
s1 ← 1; s2 ← 0;              (инициализация)
r1 ← 0; r2 ← 1;
while (r2 > 0)
{
q ← r1 / r2;

r ← r1 – q · r2;             (обновление r)
r1 ← r2; r2 ← r;

s ← s1 – q · s2;             (обновление s)
s1 ← s2; s2 ← s;

t ← t1 – q · t2;             (обновление t)
t1 ← t2; t2 ← t;
}
НОД (a, b) ← r1; s ← s1; t ← t1
```

Рисунок 4 – Блок-схема (а) и программная реализация (б) расширенного алгоритма Евклида

Расширенный алгоритм Евклида может быть использован для нахождения решения линейных диофантовых уравнений.

Линейное диофантово уравнение – это уравнение двух переменных:

$$ax + by = c. \qquad (4)$$

Требуется найти значения целых чисел для x и y, которые удовлетворяют уравнению (4). Этот тип уравнения либо не имеет решений, либо имеет бесконечное число решений.

Пусть d = НОД (a, b). Если $d \dagger c$, то линейное диофантово уравнение не имеет решения; если $d|c$, то имеется бесконечное число решений (одно из них называется частным, остальные – общими).

Алгоритм нахождения частного решения линейного диофантова уравнения включает:

– преобразование уравнения (4) к виду $a_1x + b_1y = c_1$ путем деления обеих частей уравнения (4) на d = НОД (a, b);

– нахождение s и t в равенстве $a_1s + b_1t = 1$, используя расширенный алгоритм Евклида;

– получение частных решений $x_0 = (c/d)s$ и $y_0 = (c/d)t$;

– нахождение общих решений:

$$x = x_0 + k(b/d) \quad \text{и} \quad y = y_0 - k(a/d), \tag{5}$$

где k – целое число.

1.2 Практическое задание

Условия практических заданий являются общими для всех вариантов, а конкретные исходные данные по каждому заданию определяются преподавателем дисциплины.

1.2.1 Найдите наибольший общий делитель двух положительных целых чисел a и b, используя алгоритм Евклида.

1.2.2 Используя расширенный алгоритм Евклида, найдите наибольший общий делитель двух положительных целых чисел a и b, а также значения целых чисел s и t, удовлетворяющих равенству $s \cdot a + t \cdot b = \text{НОД}(a, b)$.

1.2.3 Найдите частные и общие решения линейного диофантова уравнения $ax + by = c$, используя расширенный алгоритм Евклида.

1.2.4 По результатам выполненных расчетов заполните таблицу 1.

Таблица 1 – Результаты выполнения заданий 1.2.1–1.2.3

Задание	Пример 1	Пример 2	Пример 3
1.2.1			
1.2.2			
1.2.3			

Примечание – По заданию 1.2.1 указать НОД; по заданию 1.2.2 указать НОД, s и t; по заданию 1.2.3 указать общие решения x и y.

1.3 Содержание отчета

1 Цель практического занятия.

2 Таблица с результатами выполнения практических заданий.

3 Выводы по результатам выполнения практических заданий.

4 Ответы на контрольные вопросы.

1.4 Контрольные вопросы

1 Какие бинарные операции используются в криптографии?

2 В чем заключается сущность алгоритма Евклида?

3 Чем отличается расширенный алгоритм Евклида от простого алгоритма Евклида?

4 Каким образом расширенный алгоритм Евклида используется для нахождения решения линейных диофантовых уравнений?

5 Как свойства теории делимости могут использоваться в криптографической арифметике целых чисел?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2
ОПЕРАТОРЫ СРАВНЕНИЯ И ИНВЕРСИИ В МОДУЛЬНОЙ АРИФМЕТИКЕ

Цель: изучение основных используемых в криптографии операций и операторов в системе наименьших вычетов.

2.1 Краткие теоретические сведения

Уравнение деления, рассмотренное в практическом занятии 1, имеет два входа (a и n) и два выхода (q и r). В модульной арифметике представляет интерес только один из выходов – остаток r. Это позволяет представить изображение уравнения деления в виде бинарного оператора с двумя входами (a и n) и одним выходом r. Такой бинарный оператор называют оператором по модулю и обозначают как mod; второй вход n бинарного оператора называют модулем, а его выход r – вычетом.

На рисунке 5 показано соотношение деления по сравнению с оператором по модулю.

a и n – входы;

q и r – выходы

Рисунок 5 – Соотношение уравнения деления и оператора по модулю

Как показано на рисунке 5, оператор по модулю выбирает целое число a из множества Z и положительный модуль n и определяет неотрицательный

остаток *r*. Результатом операции по модулю *n* всегда является неотрицательное целое число в диапазоне [0, *n* – 1]. Таким образом, можно сказать, что создается набор, который в модульной арифметике понимается как система (набор) наименьших неотрицательных вычетов по модулю *n* или *Zn*.

В криптографии достаточно часто используется понятие сравнения вместо равенства [8–12]. Отображение *Z* в *Zn* не отображается «один в один»; бесконечные элементы множества *Z* могут быть отображены одним элементом *Zn*. Для указания того, что два целых числа сравнимы, используется оператор сравнения ≡. Чтобы определить значение модуля и сделать равенство правильным, к правой стороне оператора сравнения добавляется mod *n*.

Таким образом, оператор равенства отображает элемент *Z* на самого себя, а оператор сравнения – на элемент *Zn*.

Система вычетов [*a*], или [*a*]*n* – это множество целых чисел, сравнимых по модулю *n*.

Таким образом, система вычетов представляет собой набор всех целых чисел, таких, что

$$x = a(\mathrm{mod}\, n). \tag{6}$$

Например, если *n* = 5, набор наименьших неотрицательных вычетов запишется как Z_5 = {0, 1, 2, 3, 4}. В этом наборе имеется множество из пяти элементов [0], [1], [2], [3] и [4]. Все целые числа в каждом элементе дают одинаковый остаток при делении на 5, т. е. они сравнимы по модулю 5:

$$[0] = \{\ldots, -15, -10, -5, 0, 5, 10, 15, \ldots\}, \tag{7}$$

$$[1] = \{\ldots, -14, -9, -4, 1, 6, 11, 16, \ldots\}, \tag{8}$$

$$[2] = \{\ldots, -13, -8, -3, 2, 7, 12, 17, \ldots\}, \tag{9}$$

$$[3] = \{\ldots, -12, -7, -2, 3, 8, 13, 18, \ldots\}, \tag{10}$$

$$[4] = \{\ldots, -11, -6, -1, 4, 9, 14, 19, \ldots\}. \tag{11}$$

При делении на 5 все целые числа в наборе [0] дают остаток 0, в наборе [1] – остаток 1 и т. д., т. е. все целые числа в соответствующих наборах сравнимы по модулю 5.

Следует отметить, что в каждом наборе имеется один наименьший (неотрицательный) вычет, например, в наборе [0] это элемент 0, в наборе [1] – элемент 1 и т. д.

Криптография часто включает в себя решение уравнения или множества уравнений одной или более переменных с коэффициентом в *Zn*.

Далее рассмотрим базовые принципы, используемые для решения линейного уравнения с одним неизвестным. Предположим, что уравнение содержит сравнение, т. е. записано в виде

$$ax \equiv b(\mathrm{mod}\, n). \qquad (12)$$

Это уравнение может не иметь ни одного решения или иметь ограниченное число решений.

Пусть d = НОД (a, n). Если $d \nmid b$, то линейное уравнение не имеет решения; если $d | b$, то имеется d решений.

Алгоритм нахождения решения линейного уравнения включает:

– сокращение уравнения (12) путем деления обеих частей на d = НОД (a, n);

– умножение обеих сторон сокращенного уравнения на мультипликативную инверсию, чтобы найти конкретное решение x_0;

– получение общих решений:

$$x = x_0 + k(n/d), \qquad (13)$$

где $k = 0, 1, \ldots, (d - 1)$.

Рассмотренные на практическом занятии 1 бинарные операции, которые используются в криптографии в арифметике целых для множества целых чисел *Z*, могут также быть определены для набора *Zn*. При этом результат соответствующей операции должен быть отображен в *Zn* с использованием операции по модулю.

Рассмотрим основные свойства бинарных операций в системе наименьших вычетов:

$$\begin{aligned}(a+b)\bmod n &= [(a \bmod n)+(b \bmod n)]\bmod n,\\ (a-b)\bmod n &= [(a \bmod n)-(b \bmod n)]\bmod n,\\ (a\cdot b)\bmod n &= [(a \bmod n)\cdot(b \bmod n)]\bmod n.\end{aligned} \qquad (14)$$

Применение третьего свойства оператора по модулю *n* позволяет находить остаток от степеней числа 10 при делении на целое число.

Свойства модульного оператора используются, в частности, для доказательства того, что остаток от целого числа, разделенного на 3, такой же, как остаток от деления суммы его десятичных цифр. Запишем целое число как сумму его цифр, умноженных на степени числа 10:

$$a = a_n 10^n + \ldots + a_1 10^1 + a_0 10^0. \qquad (15)$$

Применим модульную операцию к двум сторонам равенства (15):

$$\begin{aligned} a \bmod 3 &= \left(a_n 10^n + \ldots + a_1 10^1 + a_0 10^0\right)\bmod 3 = \\ &= \left[\left(a_n 10^n\right)\bmod 3 + \ldots + \left(a_1 10^1\right)\bmod 3 + \left(a_0 10^0\right)\bmod 3\right]\bmod 3 = \\ &= \left[\left(a_n \bmod 3\cdot 10^n \bmod 3\right)+\ldots+\left(a_1 \bmod 3\cdot 10^1 \bmod 3\right)+\right. \\ &\quad \left.+\left(a_0 \bmod 3\cdot 10^0 \bmod 3\right)\right]\bmod 3 = \\ &= \left[\left(a_n \bmod 3\right)+\ldots+\left(a_1 \bmod 3\right)+\left(a_0 \bmod 3\right)\right]\bmod 3 = \\ &= \left(a_n+\ldots+a_1+a_0\right)\bmod 3. \end{aligned} \qquad (16)$$

В криптографии в модульной арифметике достаточно часто требуется вычислить величину, обратную заданному числу, например, аддитивную инверсию (оператор, обратный сложению) или мультипликативную инверсию (оператор, обратный умножению).

В отображении *Zn* два числа *a* и *b* аддитивно инверсны друг другу, если

$$b = n - a. \tag{17}$$

В отображении *Zn* два числа *a* и *b* мультипликативно инверсны друг другу, если

$$a \cdot b \equiv 1 \bmod n. \tag{18}$$

Может быть доказано, что некоторое число *a* имеет мультипликативную инверсию в *Zn*, если только

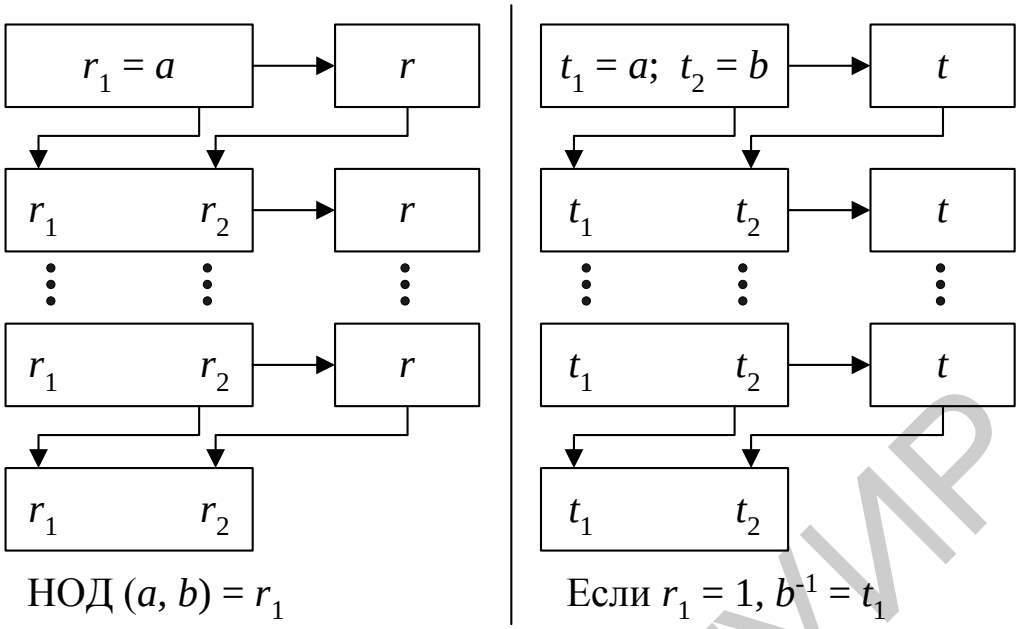
$$\text{НОД}\,(n, a) = 1. \tag{19}$$

Если условие (19) выполняется, то говорят, что *a* и *n* взаимно простые. Если условие (19) не выполняется, то мультипликативная инверсия не существует.

Расширенный алгоритм Евклида, рассмотренный на практическом занятии 1, может быть использован для поиска мультипликативной инверсии.

На рисунке 6 показан принцип реализации расширенного алгоритма Евклида для поиска мультипликативной инверсии.

а



б

```
r1 ← n; r2 ← b;
t1 ← 0; t2 ← 1;                (инициализация)

while (r2 > 0)
{
q ← r1 / r2;

r ← r1 – q · r2;               (обновление r)
r1 ← r2; r2 ← r;

t ← t1 – q · t2;               (обновление t)
t1 ← t2; t2 ← r;
}

if (r1 = 1), then b-1 ← t1
```

Рисунок 6 – Блок-схема (а) и программная реализация (б) расширенного алгоритма Евклида для поиска мультипликативной инверсии

Расширенный алгоритм Евклида позволяет найти мультипликативную инверсию числа b в Zn, когда даны n и число b и инверсия существует. Для этого необходимо заменить первое целое число a на n (модуль), затем с помощью расширенного алгоритма Евклида найти другие два целых числа s и t, при которых выполняется равенство

$$s \cdot n + t \cdot b = \text{НОД}(n, b). \tag{20}$$

Как отмечалось ранее, мультипликативная инверсия числа b существует, когда НОД (n, b) = 1, поэтому выражение (20) примет вид

$$s \cdot n + t \cdot b = 1. \tag{21}$$

Применим операции по модулю n к обеим сторонам уравнения (21):

$$\begin{aligned} (s \cdot n + t \cdot b) \bmod n &= 1 \bmod n, \\ [(s \cdot n) \bmod n + (t \cdot b) \bmod n] &= 1 \bmod n. \end{aligned} \tag{22}$$

После упрощения получаем

$$(t \cdot b) \bmod n = 1. \tag{23}$$

Таким образом, выполнение условия (23) означает, что t является мультипликативной инверсией числа b в отображении Zn.

На рисунке 7 в качестве примера показаны таблицы для сложения и умножения в отображении Z_{10}.

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

а

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

б

Рисунок 7 – Таблицы сложения (а) и умножения (б) в отображении Z_{10}

Обе таблицы симметричны по диагонали: от верхней вершины слева к нижней вершине справа. При этом можно обнаружить свойства коммутативности для сложения и умножения:

$$\begin{aligned} a+b &= b+a, \\ a\cdot b &= b\cdot a. \end{aligned} \tag{24}$$

Таблица сложения также показывает, что каждый ряд или колонка может поменяться с другим рядом или колонкой. Для таблицы умножения это неверно.

Операторы сравнения и инверсии находят широкое практическое применение в различных криптографических системах связи. В качестве примера далее рассмотрим использование этих операторов в криптосистеме RSA.

Алгоритм RSA является первым полноценным алгоритмом с открытым ключом, который может работать как в режиме шифрования данных, так и в режиме электронной цифровой подписи [13].

Надежность алгоритма основывается на трудности факторизации больших чисел и трудности вычисления дискретных логарифмов.

В криптосистеме RSA открытый ключ K_B, секретный ключ k_B, сообщение M и криптограмма C принадлежат множеству целых чисел:

$$Z_N = (0,\ 1,\ 2..., N-1), \tag{25}$$

где N – модуль, который равен

$$N = PQ, \tag{26}$$

где P и Q – случайные большие простые числа.

Для обеспечения информационной безопасности P и Q выбирают равной длины и хранят в секрете.

Таким образом, множество Z_N с операциями сложения и умножения по модулю N образует арифметику по модулю N.

Открытый ключ K_B выбирают случайным образом так, чтобы выполнялись следующие условия:

$$1 < K_B \le \varphi(N), \tag{27}$$

$$\text{НОД}\ (K_B, \varphi(N)) = 1, \tag{28}$$

где $\varphi(N)$ – phi-функция Эйлера $\varphi(n)$.

Phi-функция Эйлера $\varphi(N)$ (или тотиента Эйлера) – это мультипликативная арифметическая функция, равная количеству натуральных чисел, меньших N и взаимно простых с ним.

Phi-функция Эйлера $\varphi(N)$ позволяет найти из ряда чисел от 0 до $(N - 1)$ числа, взаимно простые с N, по следующим правилам:

– $\varphi(1) = 0$;

– $\varphi(p) = p - 1$ и $\varphi(p^x) = p^x - p^{x-1}$, если p – простое число;

– $\varphi(M \cdot N) = \varphi(M) \cdot \varphi(N)$, если M и N – взаимно простые числа.

Следовательно, для криптосистемы RSA

$$\varphi(N) = (P-1)(Q-1). \tag{29}$$

Условие (28) означает, что открытый ключ K_B и функция Эйлера $\varphi(N)$ должны быть взаимно простыми.

Далее, используя расширенный алгоритм Евклида (см. практическое занятие 1), вычисляют секретный ключ k_B, такой, что

$$k_B K_B \equiv 1 \bmod \varphi(N), \tag{30}$$

или

$$k_B = K_B^{-1} (\bmod (P-1)(Q-1)). \tag{31}$$

Преобразование шифрования в криптосистеме RSA определяет криптограмму C через пару (открытый ключ K_B, сообщение M) в соответствии со следующей формулой:

$$C = M^{K_B} \bmod N. \tag{32}$$

В качестве алгоритма быстрого вычисления значения C используют ряд последовательных возведений в квадрат целого M и умножений на M с приведением по модулю N.

Обращение функции (32), т. е. определение значения M по известным значениям C, K_B и N, практически не осуществимо при длине N не менее 1500 десятичных разрядов. Выполнить обратную задачу, т. е. задачу расшифрования криптограммы C, можно, используя пару (секретный ключ k_B, криптограмма C) по следующей формуле:

$$M = C^{k_B} \bmod N. \tag{33}$$

Процесс расшифрования можно записать так:

$$D_B[E_B(M)] = M. \tag{34}$$

где E_B – алгоритм шифрования данных;

D_B – алгоритм расшифрования данных.

Подставляя в (34) значения (32) и (33), получаем

$$\left(M^{K_B}\right)^{k_B} = M \bmod N. \qquad (35)$$

Таким образом, получатель B, который создает криптосистему RSA, защищает два параметра: секретный ключ k_B и пару чисел (P, Q), произведение которых дает значение модуля N. С другой стороны, получатель B открывает значение модуля N и открытый ключ K_B.

Противнику известны лишь значения K_B и N. Если бы он смог разложить число N на множители P и Q, то узнал бы «потайной ход» – тройку чисел (P, Q, K_B), вычислил значение функции Эйлера $\varphi(N)$ по формуле (29) и определил значение секретного ключа k_B. Однако разложение большого N на множители вычислительно не осуществимо при условии, что длина N составляет не менее 1500 десятичных разрядов, как отмечалось ранее.

Предположим, что пользователь A хочет передать пользователю B сообщение M в зашифрованном виде, используя криптосистему RSA. В таком случае пользователь A выступает в роли отправителя сообщения, а пользователь B – в роли получателя. Криптосистему RSA должен сформировать получатель сообщения, т. е. пользователь B.

На рисунке 8 показана криптографическая система связи, использующая для шифрования и расшифрования пользовательских данных алгоритм RSA. Далее для упрощения вычислений будут использоваться небольшие числа.

Пусть, например, необходимо зашифровать сообщение «CAB».

Рассмотрим последовательность действий, которую должны выполнить отправитель A (Алиса) и получатель B (Боб).

Вначале пользователь B выбирает два произвольных простых числа $P = 3$ и $Q = 11$ и вычисляет по формулам (26) и (29) соответственно значения модуля $N = 33$ и функции Эйлера $\varphi(N) = 20$.

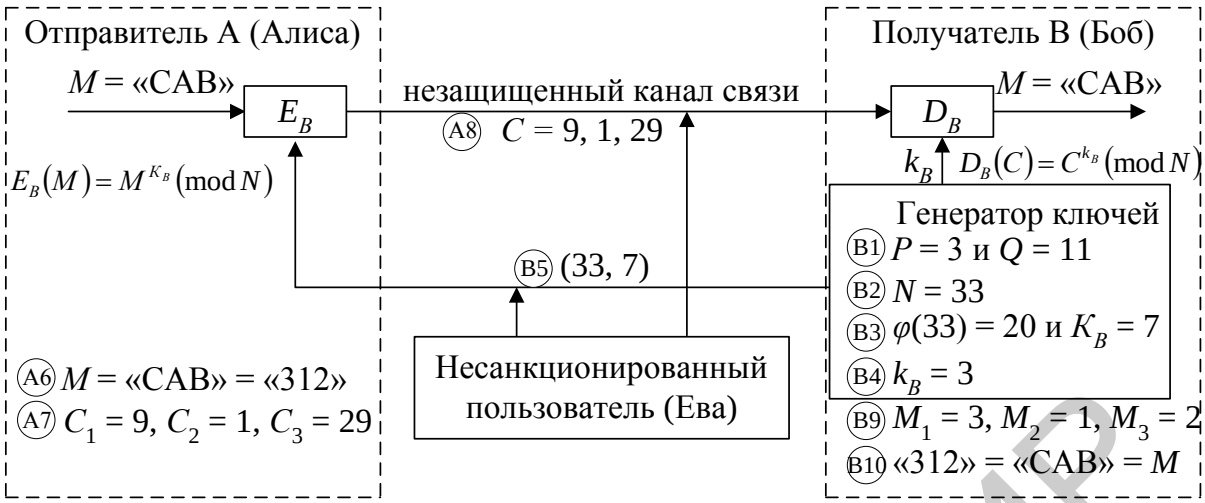


Рисунок 8 – Криптографическая система связи на базе алгоритма RSA

Затем пользователь В случайным образом выбирает значение открытого ключа K_B с учетом выполнения условий (27) и (28). Пусть, например, $K_B = 7$. После этого пользователь В вычисляет значение секретного ключа k_B, используя расширенный алгоритм Евклида при решении сравнения $k_B \equiv 7^{-1} \bmod 20$. Решение дает значение $k_B = 3$. Далее пользователь В пересылает пользователю А пару чисел (33, 7) по незащищенному каналу.

Пользователь А на основании сообщения M = «CAB» = «312», открытого ключа $K_B = 7$ и модуля $N = 33$ по формуле (32) вычисляет криптограмму $C = \{9, 1, 29\}$ и отправляет ее по незащищенному каналу связи пользователю В. Пользователь В принимает криптограмму $C = \{9, 1, 29\}$ и на основании выражения (33) расшифровывает ее, используя хранящиеся значения модуля $N = 33$ и секретного ключа $k_B = 3$. В результате пользователь В получает исходное сообщение M = «CAB» = «312».

Таким образом, в алгоритме RSA можно выделить три этапа: генерация ключей, шифрование и расшифрование.

Информационная безопасность алгоритма RSA базируется на трудности решения задачи факторизации больших чисел, являющихся произведениями двух больших простых чисел. Действительно, криптостойкость алгоритма RSA

определяется тем, что после формирования секретного ключа k_B и открытого ключа K_B «стираются» значения простых чисел P и Q, и тогда исключительно трудно определить секретный ключ k_B по открытому ключу K_B, поскольку для этого необходимо решить задачу нахождения делителей P и Q модуля N. Разложение величины N на простые множители P и Q позволяет вычислить функцию $\varphi(N)=(P-1)(Q-1)$ и затем определить секретное значение k_B, используя уравнение (31).

Другим возможным способом криптоанализа алгоритма RSA является непосредственное вычисление или подбор значения функции $\varphi(N)=(P-1)(Q-1)$. Если установлено значение $\varphi(N)$, то сомножители P и Q вычисляются достаточно просто. Пусть

$$\begin{aligned} X &= P+Q = N+1-\varphi(N), \\ Y &= (P-Q)^2 = (P+Q)^2 - 4N. \end{aligned} \tag{36}$$

Зная $\varphi(N)$, можно определить X и затем Y; зная X и Y, можно определить числа P и Q из следующих соотношений:

$$P = 1/2\left(x+\sqrt{y}\right), \quad Q = 1/2\left(x-\sqrt{y}\right). \tag{37}$$

Однако эта атака не проще задачи факторизации модуля N.

Далее рассмотрим основные виды атак, которые возможны на криптосистемы RSA.

Атака разложения на множители модуля N может быть реализована, если модуль N выбран небольшим. Первоначально авторы алгоритма RSA предлагали для вычисления модуля N выбирать простые числа P и Q случайным образом по 50 десятичных разрядов каждое. Считалось, что такие большие числа N очень трудно разложить на простые множители. Один из авторов алгоритма

RSA P. Райвест полагал, что разложение на простые множители числа из почти 130 десятичных цифр потребует более 40 квадриллионов лет машинного времени. Однако этот прогноз не оправдался из-за сравнительно быстрого прогресса компьютеров и их вычислительной мощности, а также улучшения алгоритмов факторизации.

Один из наиболее быстрых алгоритмов факторизации, известных в настоящее время, алгоритм NFS (Number Field Sieve), может выполнить факторизацию большого числа N (с числом десятичных разрядов больше 120) за число шагов, оцениваемых величиной $e^{2(\ln N)^{1/3}\left((\ln(\ln N))^{2/3}\right)}$.

В 1994 г. было факторизовано число со 129 десятичными цифрами. Это удалось осуществить математикам А. Ленстре и М. Манасси посредством организации распределенных вычислений на 1600 компьютерах, объединенных сетью, в течение 8 месяцев. По мнению А. Ленстры и М. Манасси, их работа компрометирует криптосистемы RSA и создает большую угрозу их дальнейшему применению. В настоящее время разработчикам криптоалгоритмов с открытым ключом на базе RSA приходится избегать применения чисел длиной менее 200 десятичных разрядов. Поэтому чаще применяют числа длиной 1500–3000 десятичных разрядов.

Атака с выборкой зашифрованного текста базируется на мультипликативном свойстве RSA. Предположим, отправитель А создает криптограмму $C = M^{K_B} \bmod N$ и передает ее получателю В, а также предположим, что получатель В расшифрует другой произвольный зашифрованный текст для несанкционированного пользователя. В этом случае несанкционированный пользователь может перехватить криптограмму C и определить M, последовательно выполнив следующие действия:

– выбрать случайное целое число X в отображении Z_{n*};

– вычислить число $Y = C \cdot X^{K_B} \left(\bmod N\right)$;

– передать получателю B значение Y для расшифрования и получить $Z = Y^{k_B} \bmod N$, что является выборкой зашифрованного текста;

– вычислить $M = Z \cdot X^{-1} (\bmod N)$, применив расширенный алгоритм Евклида для определения X^{-1}, поскольку

$$Z = Y^{k_B} \bmod N = \left(C \cdot X^{K_B}\right)^{k_B} \bmod N =$$
$$= C^{k_B} \cdot X^{K_B k_B} (\bmod N) = M \cdot X (\bmod N). \quad (38)$$

Для сокращения времени шифрования можно использовать короткий ключ шифрования K_B, например, выбрав второе простое число $K_B = 3$, однако это снижает информационную безопасность криптосистемы RSA и делает ее уязвимой к атаке на показатель степени шифрования K_B. В этой связи рекомендуется на практике использовать $K_B \geq 65\ 537$ или простое число, близкое к этому значению.

Примером атаки на показатель степени шифрования K_B является атака широковещательной передачи. Эта атака может быть начата, если отправитель A передает одно и то же сообщение M трем получателям с тем же самым общедоступным ключом и разными модулями N_1, N_2 и N_3:

$$C_1 = M^{K_B} \bmod N_1, C_2 = M^{K_B} \bmod N_2, C_3 = M^{K_B} \bmod N_3. \quad (39)$$

Применяя китайскую теорему об остатках к этим трем уравнениям, несанкционированный пользователь может найти уравнение вида

$$C' = M^{K_B} \bmod (N_1 \cdot N_2 \cdot N_3) \quad (40)$$

и вычислить значение C'.

К атаке на показатель степени расшифрования k_B относятся атаки раскрытого показателя степени расшифрования и малого значения показателя степени расшифрования. Доказано, что если в криптосистеме RSA показатель степени расшифрования k_B скомпрометирован, тогда для восстановления информационной безопасности криптосистемы должен быть заново сгенерирован не только секретный ключ расшифрования k_B, но и простые числа P и Q, модуль N, следовательно, требуется повторно сгенерировать открытый ключ шифрования K_B.

Атака малого значения показателя степени расшифрования k_B становится возможной, если получатель B, стремясь, например, ускорить процедуру расшифрования, выберет короткий ключ расшифрования k_B.

На практике криптосистемы RSA реализуются как аппаратным, так и программным способом. Для аппаратной реализации операций шифрования и расшифрования RSA разработаны специальные процессоры. Эти процессоры реализованы на сверхбольших интегральных схемах (СБИС), которые позволяют выполнять операции RSA, связанные с возведением больших чисел в колоссально большую степень по модулю N за относительно короткое время. Одна из самых быстрых аппаратных реализаций RSA с модулем 512 бит на сверхбольшой интегральной схеме имеет быстродействие 64 кбит/с. Лучшими из серийно выпускаемых СБИС являются процессоры фирмы CYLINK, выполняющие 1024-битовое шифрование RSA.

Программная реализация алгоритмов типа RSA значительно сложнее и менее производительна, чем реализации классических криптоалгоритмов типа DES. Вследствие сложности реализации операций модульной арифметики криптоалгоритм RSA обычно используют только для шифрования небольших объемов информации, например, для рассылки классических секретных ключей или в алгоритмах цифровой подписи, а основную часть пересылаемой информации шифруют с помощью симметричных алгоритмов.

2.2 Практическое задание

Условия практических заданий являются общими для всех вариантов, а конкретные исходные данные по каждому заданию определяются преподавателем дисциплины.

2.2.1 Найдите решение линейного уравнения, содержащего сравнение $ax \equiv b \pmod{n}$.

2.2.2 Найдите решение линейного уравнения, содержащего сравнение $ax + c \equiv b \pmod{n}$.

2.2.3 Найдите результат c бинарных операций, выполняемых над двумя целыми числами a и b в отображении Zn.

2.2.4 Найдите результат c трех бинарных операций, выполняемых над двумя целыми числами a и b в отображении Zn, предварительно применив основные свойства бинарных операций в системе наименьших вычетов.

2.2.5 Найдите результат $10^n \bmod x$, используя основные свойства бинарных операций в системе наименьших вычетов.

2.2.6 Определите аддитивную инверсию числа a в отображении Zn. Ответ представьте в виде пары значений, указав число и его аддитивную инверсию в требуемом отображении.

2.2.7 Найдите мультипликативную инверсию числа b в отображении Zn, используя расширенный алгоритм Евклида.

2.2.8 По результатам выполненных расчетов заполните таблицу 2.

Таблица 2 – Результаты выполнения заданий 2.2.1–2.2.7

Задание	Пример 1	Пример 2	Пример 3
1	2	3	4
2.2.1			
2.2.2			
2.2.3			
2.2.4			

Продолжение таблицы 2

1	2	3	4
2.2.5			
2.2.6			
2.2.7			

Примечание – По заданиям 2.2.1 и 2.2.2 указать общие решения x и число решений; по заданиям 2.2.3 и 2.2.4 указать результаты сложения, вычитания и умножения в требуемом отображении; по заданию 2.2.5 указать остаток от степеней числа 10 в требуемом отображении; по заданию 2.2.6 указать пары полученных значений (a, a^{-1})*; по заданию 2.2.7 указать* a^{-1}.

2.3 Содержание отчета

1 Цель практического занятия.

2 Таблица с результатами выполнения практических заданий.

3 Выводы по результатам выполнения практических заданий.

4 Ответы на контрольные вопросы.

2.4 Контрольные вопросы

1 Что называется системой вычетов?

2 Какой алгоритм используется для решения линейных уравнений, содержащих оператор сравнения?

3 Каким образом расширенный алгоритм Евклида может применяться для поиска мультипликативной инверсии числа в отображении *Zn*?

4 Какие свойства имеют бинарные операции в системе наименьших вычетов?

5 При каком условии мультипликативная инверсия числа в отображении *Zn* существует?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3
КРИПТОГРАФИЧЕСКИЕ ОПЕРАЦИИ
С МАТРИЦАМИ ВЫЧЕТОВ

Цель: изучение используемых в криптографии матриц и операций с матрицами вычетов.

3.1 Краткие теоретические сведения

Матрицы имеют аддитивные и мультипликативные инверсии.

Аддитивная инверсия матрицы A – это другая матрица B, такая, что

$$A + B = 0. \tag{41}$$

Следовательно, $b_{ij} = -a_{ij}$ для всех значений i и j (a_{ij} и b_{ij} – элементы матрицы A и матрицы B соответственно, расположенные в i-й строке и j-м столбце). Обычно аддитивная инверсия матрицы A обозначается как $(-A)$.

Мультипликативная инверсия определена только для квадратных матриц.

Мультипликативная инверсия квадратной матрицы A – это другая матрица B, такая, что

$$A \cdot B = B \cdot A = I. \tag{42}$$

где I – единичная матрица.

Обычно мультипликативная инверсия обозначается как A^{-1}.

Мультипликативная инверсия существует, только если ее детерминант, обозначаемый как det (A), $|A|$ или $\Delta(A)$, имеет мультипликативную инверсию в соответствующем инверсном множестве. Если целое число не имеет мультипликативной инверсии в Z, то не существует мультипликативной инверсии

матрицы в Z. Однако матрицы с реальными элементами имеют инверсии, только если $\det(A) \neq 0$.

В криптографии используются матрицы вычетов, которые могут содержать все элементы из Zn. Операции с матрицами вычетов выполняются так же, как и на матрицах целых чисел, за исключением того, что операции производятся в модульной арифметике.

Для матрицы вычетов характерно следующее свойство: матрица вычетов имеет мультипликативную инверсию, если детерминант матрицы имеет мультипликативную инверсию в Zn, т. е. если

$$\text{НОД}(\det(A), n) = 1. \quad (43)$$

Две матрицы, сравнимые по модулю n, записываются как

$$A \equiv B \bmod n, \quad (44)$$

если они имеют одинаковое число строк и столбцов и все соответствующие элементы сравнимы по модулю n.

Следовательно, $a_{ij} \equiv b_{ij} \bmod n$ для всех значений i и j.

3.2 Практическое задание

Условия практических заданий являются общими для всех вариантов, а конкретные исходные данные по каждому заданию определяются преподавателем дисциплины.

3.2.1 Определите, имеет ли матрица вычетов $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$ мультипликативную инверсию в отображении *Zn* и значение ее детерминанта.

3.2.2 Постройте матрицу *B* и найдите ее детерминант в отображении *Zn*. Для каждого элемента b_{ij} матрицы *B* выполняется условие $a_{ij}\ b_{ij} \equiv x \bmod n$, где a_{ij} – соответствующие элементы матрицы вычетов $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$ в отображении *Zn*.

3.2.3 По результатам выполненных расчетов заполните таблицу 3.

Таблица 3 – Результаты выполнения заданий 3.2.1, 3.2.2

Задание	Пример 1	Пример 2	Пример 3
3.2.1			
3.2.2			

Примечание – По заданию 3.2.1 указать det (A) и признак наличия (+) или отсутствия (–) A^{-1}; по заданию 3.2.2 указать B и det (B) в требуемом отображении.

3.3 Содержание отчета

1 Цель практического занятия.

2 Таблица с результатами выполнения практических заданий.

3 Выводы по результатам выполнения практических заданий.

4 Ответы на контрольные вопросы.

3.4 Контрольные вопросы

1 Что называется аддитивной инверсией матрицы?

2 Каким образом определяется наличие мультипликативной инверсии матрицы в соответствующем инверсном множестве?

3 Что называется матрицей вычетов?

4 В каком случае матрицы сравнимы по модулю n?

5 Какие криптографические операции выполняются с матрицами вычетов?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4
ПРОСТЫЕ ЧИСЛА В МОДУЛЬНОЙ АРИФМЕТИКЕ

Цель: изучение простых чисел и их применения в асимметрично-ключевой криптографии.

4.1 Краткие теоретические сведения

Асимметрично-ключевая криптография широко использует простые числа, которые являются положительными целыми числами, делимыми без остатка на единицу и на самих себя.

Составной объект (составное число) – это положительное целое число больше чем с двумя делителями.

Два положительных целых числа a и b являются взаимно простыми числами, если НОД (a, b) = 1 (рассмотрено в практическом занятии 2).

Для определения количества простых чисел, меньших или равных некоторому достаточно большому числу n, можно воспользоваться выражением для приблизительного расчета (Гаусс обнаружил верхний предел, Лагранж – нижний):

$$\left[n/\ln n\right] < \pi(n) < \left[n/(\ln n - 1{,}08366)\right], \qquad (45)$$

где $\pi(n)$ – количество простых чисел, меньших или равных n.

Например, $\pi(n)$ для $n = 10^6$ согласно (45) составляется диапазон значений от 72 383 до 78 543. При этом фактическое число простых чисел, равное 78 498, выражение (45) определить не позволяет.

Для установления, является ли число n простым, можно использовать следующие алгоритмы и способы:

– способ, заключающийся в проверке делимости без остатка числа всеми простыми числами, меньшими чем $\sqrt{n}$;

– алгоритм решета Эратосфена, сущность которого заключается в том, что вначале записывают все простые числа от 2 до n, определяют простые числа от 2 до $\sqrt{n}$ и последовательно исключают те числа, которые делят без остатка искомые простые числа.

Рассмотрим применение теоремы Ферма в криптографии. Первая версия теоремы Ферма говорит, что если p – простое число и a – целое число, такое, что p не является делителем a, тогда

$$a^{p-1} \equiv 1 \bmod p. \tag{46}$$

Вторая версия теоремы Ферма. Если p – простое число и a – целое число, тогда

$$a^{p} \equiv a \bmod p. \tag{47}$$

Теоремы Ферма применяются в криптографии для быстрого нахождения оператора сравнения при возведении в степень и для поиска мультипликативных инверсий, если модуль p является простым числом.

Если модуль p – простое число и число a – целое число, такое, что p не является его делителем, тогда

$$a^{-1} \bmod p = a^{p-2} \bmod p. \tag{48}$$

Выражение (48) может быть доказано, если умножить обе стороны равенства на число a и использовать первую версию малой теоремы Ферма. Следова-

тельно, выражение (48) позволяет не использовать расширенный алгоритм Евклида для нахождения мультипликативных инверсий.

Рассмотрим некоторые примеры применения теоремы Ферма в криптографии для нахождения мультипликативных инверсий: $8^{-1} \bmod 17 = 15 \bmod 17$, $5^{-1} \bmod 23 = 14 \bmod 23$.

Теорему Эйлера можно представить как обобщение малой теоремы Ферма. Модуль в теореме Ферма – простое число, модуль в теореме Эйлера – целое число. Рассмотрим две версии теоремы Эйлера:

– если a и n – взаимно простые числа, то

$$a^{\varphi(n)} \equiv 1 \bmod n; \tag{49}$$

– если $n = p \cdot q$, причем $a < n$, а k – целое число, то

$$a^{k \cdot \varphi(n)+1} \equiv a \bmod n. \tag{50}$$

Вторая версия теоремы Эйлера используется в криптографической системе, реализующей алгоритм RSA. Первая версия теоремы Эйлера применяется в криптографии для быстрого нахождения оператора сравнения при возведении в степень, например, $6^{24} \bmod 35 = 6^{\varphi(35)} \bmod 35 = 1$.

Теорема Эйлера также может использоваться для нахождения мультипликативной инверсии по простому модулю и по составному модулю, если a и n являются взаимно простыми числами:

$$a^{-1} \bmod n = a^{\varphi(n)-1} \bmod n. \tag{51}$$

Выражение (51) может быть доказано, если умножить обе стороны равенства на число a и использовать первую версию теоремы Эйлера (49).

Китайская теорема об остатках используется для решения систем уравнений с одной переменной, но различными взаимно простыми модулями:

$$\begin{cases} x \equiv a_1 (\mathrm{mod} m_1), \\ x \equiv a_2 (\mathrm{mod} m_2), \\ \quad \ldots \\ x \equiv a_k (\mathrm{mod} m_k), \end{cases} \tag{52}$$

где $a_1, a_2, \ldots a_k \in Zn$.

Система (52) имеет единственное решение, если модули m_1, m_2, … m_k являются взаимно простыми.

Решение системы уравнений выполняется в следующем порядке:

– вычисляют общий модуль:

$$M = m_1 \cdot m_2 \cdot \ldots \cdot m_k; \tag{53}$$

– определяют отношения:

$$M_1 = M / m_1, M_2 = M / m_2, \ldots M_k = M / m_k; \tag{54}$$

– используя соответствующие модули, m_1, m_2, … m_k, рассчитывают $M_1^{-1} (\mathrm{mod}\ m_1)$, $M_2^{-1} (\mathrm{mod}\ m_2)$, … $M_k^{-1} (\mathrm{mod}\ m_k)$;

– определяют решение системы уравнений (52):

$$x = \left(a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \ldots + a_k M_k M_k^{-1} \right) \mathrm{mod}\, M. \tag{55}$$

Следует отметить, что система уравнений (52) может иметь решение, даже если модули не являются взаимно простыми. Однако в криптографии в основном требуется находить решение системы с взаимно простыми модулями.

4.2 Практическое задание

Условия практических заданий являются общими для всех вариантов, а конкретные исходные данные по каждому заданию определяются преподавателем дисциплины.

4.2.1 Используя китайскую теорему об остатках, найдите решение системы уравнений $\begin{cases} x \equiv a_1 (\mathrm{mod} m_1), \\ x \equiv a_2 (\mathrm{mod} m_2), \\ x \equiv a_3 (\mathrm{mod} m_3). \end{cases}$

4.2.2 Используя китайскую теорему об остатках, найдите решение системы уравнений $\begin{cases} b_1 x \equiv a_1 (\mathrm{mod} m_1) \\ b_2 x \equiv a_2 (\mathrm{mod} m_2) \end{cases}$.

4.2.3 По результатам выполненных расчетов заполните таблицу 4.

Таблица 4 – Результаты выполнения заданий 4.2.1, 4.2.2

Задание	Пример 1	Пример 2	Пример 3
4.2.1			
4.2.2			

Примечание – Указать x в требуемом отображении.

4.3 Содержание отчета

1 Цель практического занятия.

2 Таблица с результатами выполнения практических заданий.

3 Выводы по результатам выполнения практических заданий.

4 Ответы на контрольные вопросы.

4.4 Контрольные вопросы

1 В чем заключается сущность алгоритма решета Эратосфена?

2 Какое практическое применение в криптографии находит тотиента Эйлера?

3 Для каких целей в криптографии применяются теоремы Ферма и Эйлера?

4 Что называется простыми числами и какое практическое применение они находят в криптографии?

5 Какое практическое применение в криптографии находит китайская теорема об остатках?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5
КВАДРАТИЧНОЕ СРАВНЕНИЕ
В МОДУЛЬНОЙ АРИФМЕТИКЕ

Цель: изучение применяемых в асимметрично-ключевой криптографии алгоритмов нахождения решений квадратичных сравнений по простому и по составному модулям.

5.1 Краткие теоретические сведения

Применительно к асимметричной криптографии весьма актуальным является нахождение решения квадратичного сравнения, имеющее следующую форму:

$$a_2 x^2 + a_1 x + a_0 = 0 (\mathrm{mod}\, n), \tag{56}$$

где $a_0, a_1, \ldots a_2 \in Zn.$

Вначале рассмотрим квадратичное сравнение с модулем в виде простого числа. Такое сравнение можно представить в виде

$$x^2 \equiv a (\mathrm{mod}\, p), \tag{57}$$

где p – простое число, являющееся также взаимно простым с числом a.

Уравнение (57) либо не имеет никакого решения, либо имеет только два решения. Например, уравнение $x^2 \equiv 3 \mod 11$ имеет два решения: $x_1 \equiv 5, 6 \pmod{11}$. Уравнение $x^2 \equiv 2 \mod 11$ не имеет решения.

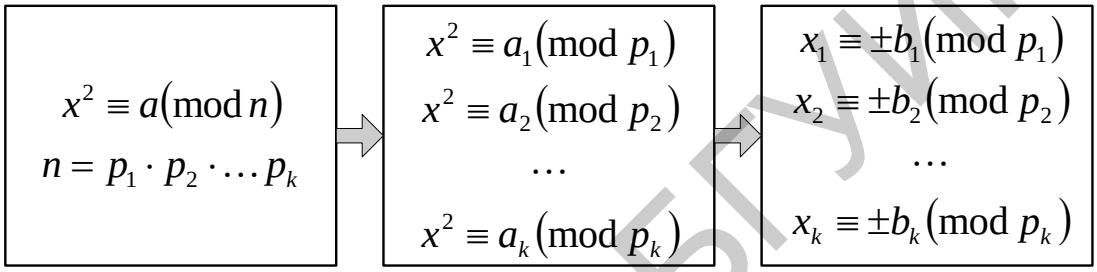
Квадратичный вычет QR – это число a в уравнении (57), если это уравнение имеет два решения.

Квадратичный невычет QNR – это число a в уравнении (57), если это уравнение не имеет решения.

В отображении Z_{p*} с $(p-1)$ элементами квадратичными вычетами являются $(p-1)/2$ элементов, а квадратичными невычетами – $(p-1)/2$ элементов.

Для определения, является ли целое число QR или QNR по модулю p, используют критерий Эйлера.

Квадратичное сравнение по составному модулю может быть приведено к решению квадратичных сравнений по модулю в виде простого числа, как показано на рисунке 9.



n – составное число;

$p_1, p_2, \ldots, p_k$ – простые числа

Рисунок 9 – Декомпозиция сравнения по составному модулю

Затем решается каждое полученное квадратичное сравнение и определяются k пар ответов. После этого из k пар ответов составляют системы уравнений, которые могут быть решены с использованием китайской теоремы об остатках, чтобы найти значения x.

5.2 Практическое задание

Условия практических заданий являются общими для всех вариантов, а конкретные исходные данные по каждому заданию определяются преподавателем дисциплины.

5.2.1 Найдите решения квадратичного сравнения $x^2 \equiv a \bmod N$, где $N = P \cdot Q$.

5.2.2 Найдите решения квадратичного сравнения $bx^2 - a \equiv 0 \bmod N$, где $N = P \cdot Q$.

5.2.3 По результатам выполненных расчетов заполните таблицу 5.

Таблица 5 – Результаты выполнения заданий 5.2.1, 5.2.2

Задание	Пример 1	Пример 2	Пример 3
5.2.1			
5.2.2			

Примечание – Указать x в требуемом отображении.

5.3 Содержание отчета

1 Цель практического занятия.

2 Таблица с результатами выполнения практических заданий.

3 Выводы по результатам выполнения практических заданий.

4 Ответы на контрольные вопросы.

5.4 Контрольные вопросы

1 Каким образом определить в квадратичном сравнении квадратичный вычет и квадратичный невычет?

2 В чем заключаются алгоритмы нахождения решений квадратичных сравнений по простому и по составному модулям?

3 Какое практическое применение в криптографии находит китайская теорема об остатках при решении квадратичных сравнений?

4 В каких криптосистемах и криптопротоколах существует необходимость решения квадратичных сравнений?

5 Какие способы применяют в асимметрично-ключевой криптографии для выбора истинного сообщения из всех возможных при расшифровании криптограммы путем решений квадратичных сравнений?

ЛИТЕРАТУРА

1 Лапонина, О. Р. Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия / О. Р. Лапонина. – М. : НОУ «Интуит», 2016. – 244 с.

2 Радько, Н. М. Основы криптографической защиты информации : учеб. пособие / Н. М. Радько, А. Н. Мокроусов. – Воронеж : ФГБОУ ВПО ВГУ, 2014. – 109 с.

3 Бабаш, А. В. Информационная безопасность. Лабораторный практикум : учеб. пособие / А. В. Бабаш, Е. К. Баранова, Ю. Н. Мельников. – М. : КНОРУС, 2012. – 136 с.

4 Введение в теоретико-числовые методы криптографии : учеб. пособие / М. М. Глухов [и др.]. – СПб. : Лань, 2011. – 400 с.

5 Голиков, В. Ф. Безопасность информации и надежность компьютерных систем : учеб. пособие. В 2 ч. Ч. 1. / В. Ф. Голиков. – Минск : БНТУ, 2010. – 86 с.

6 Стохастические методы и средства защиты информации в компьютерных системах и сетях / М. А. Иванов [и др.]. – М. : Кудиц-пресс, 2009. – 512 с.

7 Мельников, В. П. Информационная безопасность и защита информации : учеб. пособие / В. П. Мельников, С. А. Клейменов, А. М. Петраков ; под ред. С. А. Клейменова. – 3-е изд. – М. : Изд. центр «Академия», 2008. – 336 с.

8 Защита информации в компьютерных сетях. Практический курс : учеб. пособие / А. Н. Андрончик [и др.] ; под ред. Н. И. Синадского. – Екатеринбург : УГТУ-УПИ, 2008. – 248 с.

9 Жданов, О. Н. Методы и средства криптографической защиты информации : учеб. пособие / О. Н. Жданов, В. В. Золотарев. – СибГАУ : Красноярск, 2007. – 217 с.

10 Бабаш, А. В. Криптография / А. В. Бабаш, Г. П. Шангин ; под ред. А. П. Шерстюка и Э. А. Применко. – М. : СОЛОН-Пресс, 2007. – 512 с.

11 Смарт, Н. Криптография / Н. Смарт. – М. : Техносфера, 2005. – 528 с.

12 Ярочкин, В. И. Информационная безопасность : учебник / В. И. Ярочкин. – М. : Академический Проспект : Трикста, 2005. – 544 с.

13 Электронный ресурс по учебной дисциплине // Криптографическая защита информации [Электронный ресурс]. – Режим доступа : https://erud.bsuir.by/?PageID=83978=1&sortBy=21724&sortOrder=0. – Дата доступа : 18.08.2017.

Св. план 2018, поз. 43

Учебное издание

Тимофеев Александр Михайлович

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 14.03.2018. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Times».
Отпечатано на ризографе. Усл. печ. л. 2,67. Уч.-изд. л. 2,8. Тираж 30 экз. Заказ 25.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики в радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6