Министерство образования Республики Беларусь
УО «Полесский государственный университет»

М.А. РОМАНОВА

КРИПТОГРАФИЯ И ЗАЩИТА ИНФОРМАЦИИ

Учебно-методическое пособие
для студентов специальности 1-40 05 01
«Информационные системы и технологии
(по направлениям)»
всех форм обучения

Пинск
ПолесГУ
2016

УДК 004.056.5(075.8)
ББК 32.972.53я73
Р69

Р е ц е н з е н т ы:
кандидат физико-математических наук В.В. Митянок;
кандидат технических наук Ю.М. Вишняков

У т в е р ж д е н о
научно-методическим советом ПолесГУ

Романова, М.А.

Р69 Криптография и защита информации : учебно-методическое пособие / М.А. Романова. – Пинск : ПолесГУ, 2016. – 47 с.

ISBN 978-985-516-457-0

Учебно-методическое пособие содержит краткие теоретические сведения, примеры решения задач и условия для лабораторных заданий по дисциплине «Криптография и охрана коммерческой информации». Рекомендуется для использования при проведении лабораторных занятий, а также для самостоятельной подготовки.

Пособие предназначено для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» всех форм обучения.

УДК 004.056.5(075.8)
ББК 32.972.53я73

ISBN 978-985-516-457-0

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Настоящее пособие включает теоретические сведения, примеры решения задач и сами задания для выполнения лабораторных работ, предлагаемые студентам специальности 1-40 05 01 «Информационные системы и технологии» по дисциплине «Криптография и охрана коммерческой информации». Пособие состоит из трех глав.

Первая глава посвящена наивным шифрам, т.е. криптосистемам с закрытым ключом или симметричным системам шифрования.

В этой главе рассмотрены шифры перестановки: шифр «железнодорожной изгороди», «столбцовый» шифр, шифр «поворотной решётки». Также здесь рассматриваются шифры подстановки: шифр Цезаря, шифр Плейфера и шифр Виженера.

Во второй главе речь идет о шифрах с открытым ключом или асимметричных криптосистемах – современных системах шифрования. Здесь рассмотрены шифры RSA, Эль-Гамаля и Рабина.

Третья глава содержит сведения об электронной цифровой подписи. Здесь вводится понятие хеш-функции и хеш-образа, а также рассмотрены два метода постановки электронной цифровой подписи: RSA и DSA.

Заключительная глава пособия посвящена одной из основных задач криптографии – протоколу доказательства с нулевым разглашением знания. Здесь рассматриваются следующие алгоритмы: Фиата-Шамира, Гиллу-Кискатра и Шнорра.

1. КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ. СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Симметри́чные криптосисте́мы (также симметричное шифрование, симметричные шифры, системы шифрования с закрытым ключом) – способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ.

1.1. Шифрование методами перестановок

Суть шифрования методами перестановки заключается в том, что исходный (открытый) текст M делится на блоки фиксированной длины, затем символы открытого текста переставляются по определенному правилу в пределах каждого блока этого текста. Эти преобразования изменяют только порядок следования символов исходного сообщения, не изменяя самих символов.

Т.о., зашифрованный (закрытый) текст C состоит из тех же символов, что и исходный текст M, но записанных в другом порядке.

При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой стойкости шифра.

Далее более подробно рассмотрим три метода шифрования перестановками: «железнодорожной изгороди», «столбцовый» (табличный) и «поворотной решетки».

Метод «железнодорожной изгороди»

Суть метода. При использовании методом «железнодорожной изгороди» исходное сообщение преобразуется в соответствии с фигурой, напоминающей по форме зигзаг.

В этом случае символы исходного текста записываются как на **Рис. 1**, а символы зашифрованного текста считываются из полученной записи построчно.

Пример 1.1. Зашифруем этим методом фразу «Первая лабораторная работа по КиОКИ» (см. **Рис. 1**). Для удобства запишем фразу таким образом:

$$M = \text{«перваялабораторнаяработапокиоки».}$$

Теперь запишем эту фразу в виде зигзага (**Рис. 1**):

п л т р п и
е я а а о я а а о к
р а б р р а б т к о
в о н о и

Рис. 1 – Шифрование методом «железнодорожная изгородь»

Далее прочитаем эту фразу построчно сверху вниз, слева направо, и получим зашифрованный текст:

$$C = \text{«плтрпиеяааояааокрабррабтковонои».}$$

«Высоту изгороди» назовем ключом К. В приведенном выше примере $K = 4$. Для расшифровки полученного текста необходимо выполнить действия, обратные тем, что были выполнены при шифровании, и использовать тот же ключ.

«Столбцовый» метод

Суть метода. Этот метод перестановки заключается в том, что исходное сообщение записывается в таблицу построчно, а затем считывается оттуда по столбцам согласно некоторому ключу, задающему порядок считывания. Этот порядок может соответствовать порядку букв ключа в соответствии с алфавитом, а если буква встречается несколько раз, то нумерация определяется порядком следования повторяющихся букв ключевого слова.

Пример 1.2. Рассмотрим тот же открытый текст $M =$ $=$ «перваялабораторнаяработапокиокик, а ключ $K =$ $=$ «криптография».

Запишем в таблицу сначала ключ, затем пронумеруем столбцы от 1 до 12 в соответствии с порядком следования букв ключа в алфавите (см. **Рис. 2**).

Дальше построчно запишем в таблицу открытый текст:

к	р	и	п	т	о	г	р	а	ф	и	я
5	8	3	7	10	6	2	9	1	11	4	12
п	е	р	в	а	я	л	а	б	о	р	а
т	о	р	н	а	я	р	а	б	о	т	а
п	о	к	и	о	к	и					

Рис. 2 – Шифрование «столбцовым» методом

Теперь прочитаем зашифрованный текст согласно нумерации столбцов. В результате получим шифротекст:

C = ббларирркртптпяяквниеооаааааоооаа.

Если этот метод применить несколько раз с разными ключами, то задача дешифрования существенно усложнится.

Метод «поворотной решётки»

Суть метода. Решетка (трафарет), т.е. прямоугольник из клетчатой бумаги $\boldsymbol{n \times m}$ клеток, некоторые клетки в котором вырезаны, совмещается с целым прямоугольником $\boldsymbol{n \times m}$ клеток и через прорези на бумагу наносятся первые буквы текста. Затем решетку поворачивают на 90º, и через прорези записываются следующие буквы. После этого прямоугольную решетку ещё и переворачивают, а затем осуществляют ещё один поворот на 90º. В результате его вырезы полностью покрывают всю площадь прямоугольника при наложении этого трафарета-решетки на чистый лист бумаги четырьмя возможными способами, причем каждая клетка оказывается под вырезом ровно один раз.

Т.о., на бумагу наносится $\boldsymbol{n \cdot m}$ букв текста. Шифротекст получается последовательным считыванием текста согласно направлению, задаваемому ключом. Если $\boldsymbol{n = m}$, то получим квадратный трафарет, для которого переворот заменяют обычным поворотом. Решетку можно изобразить прямоугольной матрицей $\boldsymbol{n \times m}$ из нулей и единиц (единица изображает прорезь). На **Рис. 3** приведена квадратная решетка $\boldsymbol{4 \times 4}$. Рассмотрим построение этой решетки:

0	0	0	1
1	0	0	0
0	0	1	0
0	0	1	0

Рис. 3 – Решетка размерности 4×4

Построим квадрат со стороной **2** и произвольно заполним его числами 1, 2, 3, 4 (**Рис. 4**).

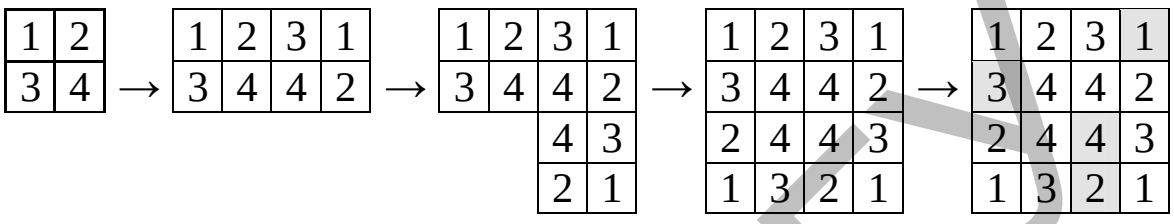

Рис. 4 – Построение решетки размерности 4×4

Копию этого квадрата повернем по часовой стрелке на 90° и присоединим к исходному справа квадрату справа. Затем копию второго квадрата повернем по часовой стрелке на 90° и присоединим ко второму квадрату снизу. А копию третьего квадрата повернем по часовой стрелке на 90° и присоединим так, чтобы получился большой квадрат $\mathbf{4 \times 4}$. Далее из большого квадрата вырезаются ровно по одной клетке с номерами, содержащей числа от 1 до 4 (эти клетки выделены серым цветом). Получили вышеупомянутую решетку $\mathbf{4 \times 4}$.

Пример 1.3. Пусть имеется открытый текст:

$M =$ «договорподписали», ключ $K =$ «шифр».

С помощью вышеуказанной решетки за пять шагов (**Рис. 5**) получаем криптограмму $C =$ «одридлппвагосоои».

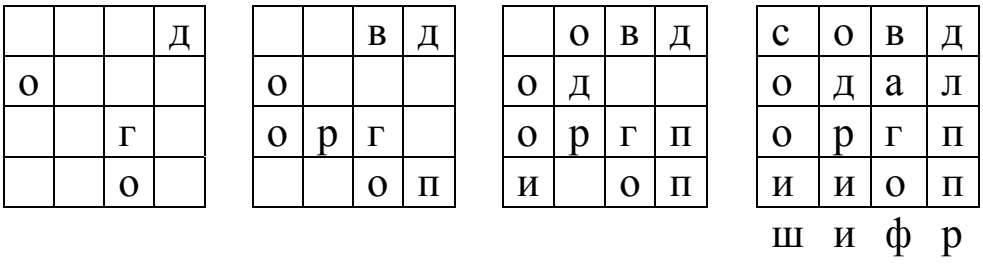

Рис. 5 – Шифрование методом «поворачивающейся решетки»

Если исходный текст не вкладывается в выбранную матрицу, то можно разбить его на блоки по 16 или менее символов.

Например, текст из 53 символов можно разбить на блоки так: 53 = 16 + 16 + 16 + 5, и каждый блок зашифровать отдельно.

1.2. Шифрование методами подстановок

Суть шифрования методом подстановки (замены) заключается в том, что символы открытого текста заменяются другими символами, взятыми из одного алфавита (одноалфавитная замена) или нескольких алфавитов (многоалфавитная). Рассмотрим три шифра подстановок: шифр Цезаря, Плейфера и Виженера.

Шифр Цезаря. Каждый символ исходного текста заменяется символом, отстоящим от исходного по алфавиту на k позиций по циклу. Подобную систему для $k = 3$ использовал Юлий Цезарь. Аналитически шифрование с помощью криптосистемы Цезаря описывается следующим равенством (1.1):

$$E_k(i) = (i + k) \text{mod n}, \qquad (1.1)$$

где n – количество букв в алфавите;

mod – означает остаток отделения. Для английского языка n = 26, для русского – 33, для ASCII-кодов – 256.

Алгоритм расшифровывания имеет вид (1.2):

$$D_k(i) = (i + n - k) \text{ mod n}. \qquad (1.2)$$

Пример 1.4. Зашифруем слово $M =$ «$example$» с ключом $k = 3$. Сначала каждой букве латинского алфавита сопоставим номер от 0 до 25. Тогда $M = (4,\ 23,\ 0,\ 12,\ 15,\ 11,\ 4)$. В соответствии с выражением (1.1) получим шифротекст $C = (7,\ 0,\ 3,\ 15,\ 18,\ 14,\ 7)$. Сопоставив по номерам буквы того же алфавита, окончательно получим $C =$ «$hadpsoh$». Расшифровать можно аналогично, используя формулу (1.2).

Замечание 1.1. Рассмотренный выше шифр простой замены легко взламывается с помощью криптографических атак, основанных на анализе частот появления символов в шифротексте. Рассмотрим шифр Плейфера, который обеспечивает защиту от таких атак.

Шифр Плейфера. *Шифр Плейфера* – наиболее известный биграммный шифр замены. Суть полиграммных алгоритмов состоит в том, что одновременно шифруется не один, а сразу несколько символов, что также позволяет видоизменить частотные зависимости, характерные для исходных текстов. При использовании шифра Плейфера кодируются сразу два символа. Основой данного шифра является шифрующая матрица со случайно расположенными буквами алфавита.

Результат получится таким (**Рис. 6**):

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

Рис. 6 – Шифрующая таблица Плейфера

Такая таблица представляет собой сеансовый ключ. Для удобства запоминания шифрующей таблицы обычно используется ключевое слово, которое определяет построение таблицы. Для случая английского языка шифрующая матрица задается таблицей, например, размером 5×5, состоящей из 25 позиций с символами алфавита английского языка (позиция для символа «J» соответствует позиции для символа «I»). Возьмем ключевое слово «CIPHERTEXT». Удалим буквы, встречающиеся более одного раза. Получим слово «CIPHERTX». Запишем слово в таблицу, дополнив остальными буквами алфавита. Процедура шифрования следующая. Сначала исходный текст M разбивается на пары символов (биграммы) $M = \{m_1m_2;\ m_3m_4;\ \dots\}$, после чего полученные биграммы $m_1m_2,\ m_3m_4, \dots$ открытого текста M преобразуются с помощью шифрующей таблицы

в последовательность биграмм c_1c_2, c_3c_4, ... шифротекста C по следующим *основным правилам*:

1. Если буквы биграммы исходного текста m_i и m_{i+1} находятся в одной и той же строке шифрующей матрицы, то c_i и c_{i+1} представляют собой два символа справа от m_i и m_{i+1} соответственно. Здесь первый столбец матрицы используется как столбец справа по отношении к последнему столбцу. Например, если $m_im_{i+1} = XB$, то $c_ic_{i+1} = AR$.

2. Если символы биграммы m_i и m_{i+1} находятся в одном и том же столбце, то c_i и c_{i+1} принимают значения символов ниже m_i и m_{i+1} соответственно. Строкой ниже последней считается первая строка. Например, если $m_im_{i+1} = RU$, то $c_ic_{i+1} = DC$.

3. Если в одной биграмме символы m_i и m_{i+1} находятся в различных строках и столбцах, то символы c_i и c_{i+1} соответствуют двум другим углам прямоугольника, имеющего m_i и m_{i+1} в качестве двух противолежащих углов. При этом c_i, находится в той же строке, что и m_i, а c_{i+1} находится в той же строке, что и m_{i+1}. Например, если $m_im_{i+1} = CK$, то $c_ic_{i+1} = HD$ (**Рис. 7**):

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

Рис. 7 – Шифрование алгоритмом Плейфера

Замечание 1.2. В любой биграмме должно выполняться условие $m_i \neq m_{i+1}$, т.е. биграмма не может состоять из двух одинаковых символов. Для этого перед шифрованием все биграммы исходного текста просматриваются, и, если нужно, текст незначительно меняют – специально подготавливают. Т.е., если всё же биграмма содержит два одинаковых символа, то между ними нужно вставить любой пустой символ («X», т.п.), чтобы устранить равенство $m_i = m_{i+1}$. Например, если m_im_{i+1} = SS, тогда $m_im_{i+1}m_{i+2}$ = SXS и, соответствен-

но, $c_i c_{i+1}$ = BO. Вторая буква S будет относиться к следующей биграмме.

Замечание 1.3. Если подготовленный исходный текст, согласно ***Замечанию 1.2***, имеет нечетное число знаков, пустой символ добавляется в конец текста для получения четного числа символов исходного текста.

Например, для исходного текста *M* = CRYPTOGRAPHY в результате шифрования по алгоритму Плейфера, используя матрицу, приведенную на **Рис. 6**, в качестве ключа получим шифротекст C = RDHWXNXDHXAH.

Замечание 1.4. Шифрование биграммами существенно повышает стойкость шифров к взлому, однако частотные свойства распределения биграмм по-прежнему являются ключом для злоумышленника.

Это обстоятельство повлекло возникновение модифицированного алгоритма Плейфера с четырьмя матрицами, который решает эту проблему.

Шифр Виженера. *Шифр Виженера* – шифр, который является шифром многоалфавитной подстановки и использует развитие идеи Цезаря. В таблице Виженера каждая строка представляет собой циклически сдвинутую на один символ предыдущую строку таблицы таким образом, что каждая строка по своей сути является таблицей подстановки шифратора Цезаря для конкретного значения ключа. Верхняя строка таблицы Виженера используется для задания символов исходных текстов, а левый столбец – для задания символов криптографического ключа. При шифровании исходного сообщения его записывают в строку, а под ним ключевое слово либо фразу. Если ключ оказался короче исходного текста, то его циклически повторяют необходимое число раз. На каждом шаге шифрования в верхней строке таблицы Виженера находят очередную букву исходного текста, а в левом столбце очередное значение символа ключа. В результате очередная буква шифротекста находится на пересечении столбца, определенного символом исходного текста и строки, соответствующей строке символа ключа.

	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
a	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	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
c	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	b
d	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
e	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	d
f	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
g	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
h	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
i	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
j	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
k	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
l	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
m	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
n	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
o	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
p	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
r	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
s	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
t	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
u	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
v	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
w	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
x	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
z	z	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

Рис. 8 – Таблица Виженера для английского языка

При шифровании слова $M = VIGENERE$ по методу Виженера для ключа $K = LEMON$ предварительно исходный текст и ключевое слово запишем в виде двух строк, третья строка будет шифротекстом (**Рис. 9**):

M = V I G E N E R E
K = L E M O N L E M
C = G M S S A P V Q

Рис. 9 – Шифрование методом Виженера

Тогда первая буква исходного текста «V» определяет 22-й столбец таблицы Виженера, а буква «L» ключа – 12-ю строку таблицы, на пересечении которых находится первый символ шифротекста «G». Аналогично для остальных букв.

Т.о., получим шифротекст C = GMSSAPVQ.

Задание для выполнения лабораторной работы № 1

1. Изучить теоретический материал по лабораторной работе.
2. Реализовать шифраторы и дешифраторы на основе одного из трех рассмотренных перестановочных методов по вариантам в соответствии с таблицей.
3. Реализовать шифратор и дешифратор на основе одного из подстановочных методов согласно варианту по **Таблице 1.1**.

Таблица 1.1. – Симметричные шифры

Вариант	Шифр перестановки	Шифр подстановки
№ 1	«железнодорожной изгороди»	Плейфера
№ 2	«столбцовый» шифр	Цезаря
№ 3	«поворотной» решетки	Виженера

4. Сформировать отчет о проделанной работе.

2. КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

Первые *криптографические системы с открытым ключом*, или *асимметричные криптосистемы*, появились в конце 1970-х гг. От классических симметричных алгоритмов они отличаются тем, что для шифрования данных используется один ключ (обычно его называют «открытый»), а для расшифровывания другой, секретный («закрытый») ключ.

Американские криптографы У. Диффи и М. Хеллман, которые впервые предложили и описали криптосистему с открытым ключом, определили следующие требования к криптосистемам:

1. Должно быть вычислительно легко создать пары «открытый ключ (K_O) – секретный (закрытый) ключ (K_C)».
2. Должно быть вычислительно легко при наличии открытого ключа зашифровать сообщение M, т.е. создать соответствующее зашифрованное сообщение $C = E_{K_O}[M]$.
3. Должно быть вычислительно легко дешифровать сообщение с использованием секретного ключа: $M = D_{K_C}[C] = = D_{K_C}[E_{K_O}[M]]$.
4. Должно быть вычислительно невозможно, зная открытый ключ K_O, определить секретный ключ K_C.
5. Должно быть вычислительно невозможно восстановить исходное сообщение M, зная открытый ключ K_O и зашифрованное сообщение C.

Можно добавить шестое требование, хотя оно выполняется не для всех алгоритмов с открытым ключом:

6. Возможность применения шифрующих и дешифрующих функций в любом порядке, т.е.:

$$M = E_{K_O}[D_{K_C}[M]] = D_{K_C}[E_{K_O}[M]].$$

Т.о., данные, зашифрованные открытым ключом, можно расшифровать только секретным ключом. Следовательно, открытый ключ может распространяться через обычные ком-

муникационные сети и другие открытые каналы, что устраняет главный недостаток стандартных криптографических алгоритмов: необходимость использовать специальные каналы связи для распределения ключей.

2.1. Криптосистема RSA

В настоящее время лучшим и наиболее популярным криптографическим алгоритмом с открытым ключом считается RSA, название которого получено по именам его создателей – Rivest, Shamir и Adelman.

Наиболее важной частью алгоритма RSA, как и других алгоритмов с открытым ключом, является процесс создания пары – открытый / секретный ключ. Далее нам понадобится следующее понятие.

Функция Эйлера $\varphi(x)$ для числа x – это количество натуральных чисел, меньших x, и взаимно простых с x.

Полезно знать некоторые частные случаи для вычисления функции Эйлера:

а) если x – простое ($x \epsilon P$), то $\varphi(x) = x - 1$;

б) если $x = p \cdot q$, где $p, q \epsilon P$, то $\varphi(x) = (p-1) \cdot (q-1)$;

в) если известна факторизация числа

$$x = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k},$$

где $p_1, p_2, \ldots, p_k \epsilon P$, то

$$\varphi(x) = x \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdot \ldots \cdot \left(1 - \frac{1}{p_k}\right). \quad (2.1)$$

В алгоритме RSA процесс генерации ключей следующий.

Алгоритм генерации ключей

1. Случайным образом выбираются два секретных простых числа p и q заданного размера (например, 1 024 бита каждое).
2. Вычисляется их произведение $n = p * q$, называемое модулем.
3. Вычисляется функция Эйлера $\varphi(n)$.

4. Выбирается целое значение переменной e такое, что $1 < e < \varphi(n)$ и НОД$(e, \varphi(n)) = 1$.

5. Вычисляется значение секретного числа d, которое должно удовлетворять условию $(e \cdot d) \bmod \varphi(n) = 1$ (т.е. d является мультипликативной инверсной по модулю $\varphi(n)$ для элемента e).

Т.о., открытым ключом K_O является пара значений $(e,\ n)$, а секретным ключом $K_C - (d,\ n)$.

Процесс шифрования

Перед шифрованием исходное сообщение M необходимо преобразовать в набор чисел m_1, m_2, ..., где $m_i \in [0; n-1]$.

Сам процесс шифрования открытым ключом $K_O = (e,\ n)$ чисел $m_1,\ m_2$, ... происходит согласно формуле:

$$c_i = (m_i{}^e) \bmod n, \qquad (2.2)$$

где последовательность чисел $C = c_1,\ c_2$, ... представляет собой шифротекст.

Процесс расшифровки

Чтобы расшифровать эти данные секретным ключом $K_C = (d, n)$, необходимо выполнить следующие вычисления:

$$m_i = (\mathrm{c}_i{}^d) \bmod n. \qquad (2.3)$$

В результате будет получено множество чисел m_i, которые представляют собой исходный текст.

Пример 2.1. С помощью метода RSA зашифровать сообщение $M =$ «DOG» и расшифровать его.

Сначала получим ключи:

1. Выберем $p = 5$, $q = 11$.
2. Вычислим $n = 5 \cdot 11 = 55$.
3. Вычислим $\varphi(n) = (p-1) \cdot (q-1) = 4 \cdot 10 = 40$.
4. Выберем открытое число *e,* которое является взаимно простым с $\varphi(n) = 40$, например, $e = 17$.
5. На основе e и $\varphi(r)$ вычислим закрытое число *d.* Для этого можно использовать расширенный алгоритма Евклида (**Рис. 10**), где $a = 40$, $b = 17$.

```
EUCLIDEX(a; b);
 begin
   d0:=a; d1:=b; x0:=1; x1:=0;
   y0:=0; y1:=1;
   while d1>1 do
      begin
        q:=d0 div d1;
        d2:=d0 mod d1;
        x2:=x0-q*x1;
        y2:=y0-q*y1;
        d0:=d1; d1:=d2; x0:=x1;
        x1:=x2; y0:=yl; y1:=y2;
      end;
   return (x1; y1; d1);
 end;
```

Рис. 10 – Алгоритм Евклида

Расширенный алгоритм Евклида позволяет вычислить числа x_1 и y_1, для которых выполняется равенство:

$$x_1 * a + y_1 * b = d_1,$$

где $d_1 = \text{НОД}\ (a, b)$.

Если a и b взаимно простые и $a > b$, то y_1 является мультипликативным инверсным по модулю a для b, т.е.:

$$(y_1 \cdot b) \bmod a = 1.$$

Используя данный алгоритм, можно вычислить d, положив $a = \varphi(n) = 40$ и $b = e = 17$. Если значение y_1 получилось отрицательным (как в нашем случае: $y_1 = -7$), то для получения корректного значения d необходимо добавить к y_1 значение a (= $\varphi(n)$).

В нашем случае имеем:

$$d = y_1 + \varphi(n) = -7 + 40 = 33.$$

Действительно, равенство $(33 \cdot 17) \bmod 40 = 1$ верно.

Итак, открытым ключом K_O является пара значений $(17,\ 55)$, а секретным ключом $K_C - (33,\ 55)$.

Вернемся к ***примеру***. Сначала представим шифруемое сообщение как набор целых чисел, например, в диапазоне 3...28 (для английского языка).

Т.о., пусть букве «A» соответствует число 3, «B» – 4, «C» – 5 и т.д. Тогда сообщению $M = «DOG»$ соответствуют числа $m_1,\ m_2,\ m_3 = \{6, 17, 9\}$. Зашифруем это сообщение {6, 17, 9}, используя открытый ключ $K_O = (17,\ 55)$ и формулу (2.1):

$$c_1 = ({m_1}^e) \bmod n\ = (6^{17}) \bmod 55 =$$
$$= 16926659444736 \bmod 55 = 41,$$

$$c_2 = (\mathrm{m_2}^{\mathrm{e}}) \bmod n = (17^{17}) \bmod 55 =$$
$$= 827240261886336764177 \bmod 55 = 52,$$

$$c_3 = ({m_3}^e) \bmod n = (9^{17}) \bmod 55 =$$
$$= 1667718169966656 9 \bmod 55 = 4.$$

Расшифруем полученный шифротекст C = {41, 52, 4} с помощью секретного ключа K_C = (33, 55) и формулы (2.3), получим:

$$m_1 = (\mathrm{c_1}^{\mathrm{d}}) \bmod \mathrm{n}\ = (41^{33}) \bmod 55\ = 6,$$

$$m_2 = \left(\mathrm{c_2}^{\mathrm{d}}\right) \bmod \mathrm{n}\ = (52^{33}) \bmod 55\ = 17,$$

$$m_3 = (\mathrm{c_3}^{\mathrm{d}}) \bmod \mathrm{n}\ = (4^{33}) \bmod 55\ = 9.$$

Т.о., в результате расшифровки сообщения получены исходные числа $\{6, 17, 9\}$, а, следовательно, и исходное сообщение $M = «DOG»$.

Важное замечание. Для возведения в степень по модулю $x = a^z \bmod n$, когда показатель степени есть достаточно большое число, можно использовать алгоритм быстрого возведения в степень по модулю (см. **Рис. 11**).

Криптостойкость алгоритма RSA основывается на двух математических трудно решаемых задачах, для которых не существует эффективного способа решения.

```
function fast_exp(a,z,n);
 begin
   a1:=a; z1:=z;
   x:=1;
   while z1<>0 do
    begin
      while (z1 mod 2)=0 do
          begin
              z1:=z1 div 2;
              a1:=(a1*a1) mod n
           end;
           z1:=z1-1;
           x:=(x*a1) mod n
      end;
    fast_exp:=x
  end;
```

Рис. 11 – Алгоритм быстрого возведения в степень (функция возвращает числовое значение по формуле $a^z \bmod n$)

Одна из них заключается в том, что невозможно вычислить исходный текст из шифротекста, т.к. для этого надо извлечь корень степени e по модулю числа n (найти дискретный логарифм). Данную задачу в настоящее время невозможно решить за полиномиальное время. Вторая задача заключается в том, что практически невозможно найти секретный ключ, зная открытый, поскольку для этого необходимо решить линейное сравнение:

$$(e \cdot d) \bmod \varphi(n) = 1.$$

Для его решения нужно знать разложение числа n на простые множители p и q. Задача разложения на множители (или задача факторизации числа) в настоящее время также не имеет эффективного (полиномиального) решения. Однако пока не было доказано и то, что не существует эффективного алгоритма решения данной задачи. Если же взять достаточно большое число n (более чем 2 048 бит), то для разложения такого числа на простые множители на обычном ПК может понадобиться несколько лет.

2.2. Криптосистема Эль-Гамаля

Эта криптосистема с открытым ключом основана на трудности вычисления дискретных логарифмов в конечном поле. Схема была предложена египетским криптографом Тахером Эль-Гамалем в 1985 г. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94).

Далее нам понадобится следующее понятие – понятие *первообразного корня по модулю*.

Первообразный корень по модулю p – это целое число g такое, что:

$$g^{\varphi(p)} = 1 \bmod p,$$

и

$$g^k \neq 1 \bmod p,$$

если $k \in [1; \varphi(p) - 1]$,

где $\varphi(p)$ – функция Эйлера (см. **п. 2.1**).

Другими словами, первообразный корень – это образующий элемент мультипликативной группы кольца вычетов по модулю p. Как и для алгоритма RSA, для криптосистемы Эль-Гамаля необходимо перед шифрованием вычислить пару «открытый / секретный ключ».

Это происходит по следующему алгоритму.

Алгоритм генерации ключей

1. Генерируется случайное простое число p.
2. Выбирается произвольное натуральное число g, являющееся первообразным корнем по модулю p.
3. Выбирается случайное целое число x такое, что $x \in [2; p - 2]$.
4. Вычисляется $y = g^x \bmod p$.

Т.о., открытым ключом K_O является тройка чисел $(p,\ g,\ y)$, секретным ключом K_C – число x.

Процесс шифрования

Число $m \in [0, p-1]$ шифруется так:

1. Выбирается случайное секретное целое число $k \in (1, p-1)$ взаимнопростое с $p-1$.
2. Вычисляется два значения a и b:

$$a = g^k \bmod p,$$
$$b = (y^k m) \bmod p. \tag{2.4}$$

где число m является исходным сообщением, а пара чисел (a, b) – шифротекстом.

Замечание. Т.о., каждому символу исходного сообщения соответствует два символа шифротекста, т.е. полученный шифротекст в два раза длиннее исходного сообщения. Также следует отметить, что для разных чисел m_1 и m_2 следует использовать и разные значения случайного k. Например, если для двух чисел m_1 и m_2 было использовано одно и то же значение k, то получим, что $\frac{b_1}{b_2} = \frac{m_1}{m_2}$, и значение m_2 может быть легко вычислено злоумышленником при известном m_1.

Процесс расшифровки

Зная секретный ключ x, исходное сообщение можно вычислить из шифротекста (a, b) по формуле:

$$m = (b \cdot a^{-x}) \bmod p = \left(b \cdot a^{x \cdot (\varphi(p)-1)}\right) \bmod p =$$
$$= (b \cdot a^{x \cdot (p-2)}) \bmod p. \tag{2.5}$$

Пример 2.2. Пусть дано сообщение $m = 5$. Зашифруем и расшифруем его с помощью алгоритма Эль-Гамаля.

Для начала сгенерируем ключи:

1. Выберем простое число $p = 11$.
2. Для $p = 11$, найдем число g, которое являлось первообразным корнем по модулю p. Например, $g = 6$.
3. В качестве закрытого ключа x возьмем, например, число 4.
4. Вычисляем $y = g^x \bmod p = 6^4 \bmod 11 = 9$. Открытым ключом является $K_O = (11,6,9)$, закрытым ключом K_C –

число 4. Далее зашифруем сообщение $m = 5$, для этого сгенерируем случайное число $k = 7$, взаимно простое с $p - 1$ и вычислим по формуле (2.3) значения a *и* b:

$$a = g^k \bmod p = 6^7 \bmod 11 = 8,$$

$$b = (y^k m) \bmod p \ = \ (9^7 \cdot 5) \bmod 11 = 9.$$

На выходе получаем шифротекст $(8, 9)$. Вычислим исходное сообщение из шифротекста (a, b) по формуле (2.4):

$$m = (ba^{-x}) \bmod p = (9 \cdot 8^{-4}) \bmod 11 =$$
$$= (9 \cdot 8^{4\cdot(11-2)}) \bmod 11 = (9 \cdot 8^{36}) \bmod 11 \ = 5.$$

Замечание. Криптостойкость криптосистемы Эль-Гамаля основана на том, что за полиномиальное время невозможно вычислить закрытый ключ, зная открытый, т.к. для этого необходимо вычислить дискретный логарифм, т.е. по известным значениям p, g и y вычислить x, который бы удовлетворял сравнению:

$$y \equiv g^x \bmod p.$$

2.3. Криптосистема Рабина

Криптосистема, разработанная М. Рабином, подобно RSA основывается на трудной проблеме факторизации больших целых чисел или на проблеме извлечения квадратного корня по модулю составного числа $n = p \cdot q$.

Алгоритм генерации ключей

Генерация ключей в криптосистеме Рабина происходит следующим образом:

1. Выбираются два разных случайных простых числа p и q таких, что $p \approx q$. При этом они должны удовлетворять условию $p \equiv q \equiv 3 \mod 4$. Выполнение данного условия необходимо для упрощения вычисления квадратного корня по модулю p и q при расшифровке сообщения.
2. Вычисляется $n = p \cdot q$.
3. Выбирается случайное число $b < n$.

Т.о., открытым ключом будет $K_O = (n, b)$, секретным – $K_C = (p, q)$.

Процесс шифрования

Для получения шифротекста с необходимо сообщение M (символ) преобразовать в число и выполнить вычисления по формуле:

$$c = (m(m + b)) \bmod n. \qquad (2.6)$$

Шифрование в алгоритме Рабина происходит намного быстрее, чем в других криптосистемах с открытым ключом.

Процесс расшифровки

Для расшифровки же нужно решить квадратное уравнение вида:

$$m^2 + b \cdot m - c = 0 \ (\bmod n).$$

Как известно, общее решение такого уравнения будет иметь вид:

$$m = \frac{-b+\sqrt{D}}{2} \ (\bmod n), \qquad (2.7)$$

где $$D = (b^2 + 4c) \ (\bmod n).$$

Вычислить квадратный корень из D по модулю числа $n = p \cdot q$ можно с помощью китайской теоремы об остатках, для чего необходимо знать закрытые ключи p и q. Вычислив $\sqrt{D}$, получим четыре результата d_1, d_2, d_3, d_4, и, подставив в (2.7), получим m_1, m_2, m_3, m_4.

Замечание. Основным недостатком криптосистемы Рабина является то, что неизвестно, который из четырех результатов m_1, m_2, m_3, m_4 равен исходному m. Если сообщение написано по-русски, выбрать правильное m нетрудно. С другой стороны, если сообщение является потоком случайных битов, способа определить, какое m правильное, практически нет.

Пример 2.3. С помощью криптосистемы Рабина зашифруем и расшифруем букву «Д». Т.к. эта буква является пятой буквой алфавита, то $m = 5$.

Сгенерируем ключи:

1. Выберем два простых числа $p = 23$ и $q = 11$. При этом $23 \bmod 4 = 3$ и $11 \bmod 4 = 3$, т.е. условие $p \equiv q \equiv 3 \bmod 4$ выполняется.

2. Вычислим наш открытый ключ:

$$n = p \cdot q = 23 \cdot 11 = 253.$$

3. Выберем случайное число b. Например,

$$b = 131 \ (b < 253).$$

Далее зашифруем сообщение $n = 5$. Для этого вычислим по формуле (2.6):

$$c = (m \cdot (m + b)) \bmod n = (5 \cdot (5 + 131)) \bmod 253 = 174.$$

Т.о., мы получили шифротекст $c = 174$. Для расшифровки сначала вычислим D по формуле (2.7):

$$D = (b^2 + 4c) \bmod n = (131^2 + 4 \cdot 174) \bmod 253 = 147.$$

Далее для вычисления квадратного корня из $D = 185$ по модулю составного числа $n = p \cdot q$ выполним следующие действия:

1. Вычислим s и r по формулам:

$$s = \sqrt{D} \bmod p = D^{\frac{p+1}{4}} \bmod p = 147^{\frac{23+1}{4}} \bmod 23 = \\ = 147^6 \bmod 23 = 3, \qquad (2.8)$$

$$r = \sqrt{D} \bmod q = D^{\frac{q+1}{4}} \bmod q = 147^{\frac{11+1}{4}} \bmod 11 = \\ = 147^3 \bmod 11 = 9. \qquad (2.9)$$

2. Вычислим значения y_p и y_q таких, что $y_p \cdot p + y_q \cdot q = 1$ по расширенному алгоритму Евклида (см. **п. 2.1**). Для этого примем $a = 23$, $b = 11$. Тогда на выходе получим, что:

$$y_p = x_1 = 1, y_q = y_1 = -2.$$

3. Далее вычисляем квадратные корни из D по модулю n по формулам:

$$d_{1,2} = \pm(y_p \cdot p \cdot r + y_q \cdot q \cdot s) \bmod n = \\ = \pm(1 \cdot 23 \cdot 9 + (-2) \cdot 11 \cdot 3) \bmod 253 = \\ = \pm(207 - 66) \bmod 253 = \pm 141 \bmod 253,$$

$$d_{3,4} = \pm\left(y_p \cdot p \cdot r - y_q \cdot q \cdot s\right) \bmod n =$$
$$= \pm\,(1 \cdot 23 \cdot 9 - (-2) \cdot 11 \cdot 3) \bmod 253 =$$
$$= \pm(207 + 66) \bmod 253 = \pm 273 \bmod 253 = \pm 20 \bmod 253.$$

Тогда получим, что: $d_1 = 141$, $d_2 = 253 - 141 = 112$, $d_3 = 20$, $d_4 = 253 - 20 = 233$. При подстановке всех значений в (2.7) получаем числа:

$$m_1 = \frac{-b+d_1}{2} \bmod n = \frac{-131+141}{2} \bmod 253 = 5,$$

$$m_2 = \frac{-b+d_2}{2} \bmod n = 243{,}5,$$

$$m_3 = 197{,}5, \quad m_4 = 51.$$

Как видим, первое из найденных значений соответствует исходному сообщению $m = 5$.

Задание дли выполнения лабораторной работы № 2

Таблица 2.1. – Асимметричные шифры

Вариант	Задание
№ 1	Реализовать шифратор и расшифровщик алгоритма RSA, используя алгоритм быстрого возведения в степень. А также реализовать вычисление пары открытый / секретный ключ при данных значениях p, q и e, используя расширенный алгоритм Евклида
№ 2	Реализовать шифратор и расшифровщик алгоритма Эль-Гамаля, используя алгоритм быстрого возведения в степень. А также реализовать вычисление открытого ключа g при данном значении p, используя один из алгоритмов нахождения первообразного корня по модулю
№ 3	Реализовать шифратор и расшифровщик по алгоритму Рабина, используя расширенный алгоритм Евклида и алгоритм быстрого возведения в степень при расшифровке

1. Изучить теоретический материал по лабораторной работе.

2. Выполнить задание согласно своему варианту (см. **Таблицу 2.1**).

3. Сформировать отчет о проделанной работе.

3. ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

3.1. Хеш-функция (функция хеширования)

Хеширование (иногда **хэширование – англ. *hashing*)** – преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины.

Такие преобразования также называются **хеш-функциями** или **функциями свёртки**, а их результаты называют хешем, хеш-образом или дайджестом сообщения (англ. *message*, *digest*).

Хорошая хеш-функция должна удовлетворять следующим условиям:

1) чувствительность к любым изменениям входной последовательности M (даже изменение регистра одного символа полностью меняет вид хеш-образа);
2) её можно применить к сообщению любой длины;
3) хеш-образ имеет фиксированную длину;
4) зная хеш-образ, невозможно восстановить исходное сообщение либо сообщение с таким же хеш-образом;
5) вычислительно невозможно найти два сообщения с одинаковыми хеш-образами;
6) вероятность возникновения коллизий, т.е. когда для разных сообщений значения их хеш-образов совпадают, должна быть чрезвычайно мала, но не равна нулю.

При построении хеш-образа исходное сообщение разбивается на блоки-символы m_i и обрабатывается последовательно по формуле:

$$H_i = f(H_{i-1}, m_i). \qquad (3.1)$$

Т.о., хеш-функция подобна пирамидке, т.е. использует хеш-образ предыдущего символа, чтобы сформировать хеш-образ следующего за ним символа. Хеш-значение, вычисленное в результате обработки последнего символа сообщения, объявляется хеш-образом всего сообщения.

В качестве примера рассмотрим упрощенный вариант хеш-функции следующего вида:

$$H_i = (H_{i-1} + m_i)^2 \bmod n, \qquad (3.2)$$

где $n = p \cdot q$,

p и q – большие простые числа;

H_O – произвольное начальное значение;

m_i – i-й блок сообщения $M = \{m_1, m_2, \dots, m_k\}$.

Пример 3.1. Вычислим хеш-образ для строки «ПолесГУ» с помощью хеш-функции (3.2). Для перехода от символов к числовым значениям будем использовать естественное соответствие 'А' – 1, 'Б' – 2, 'В' – 3, ..., 'Я' – 33. Тогда сообщение M примет вид $M = \{17, 16, 13, 6, 19, 4, 21\}$. Выберем два простых числа, например, $p = 17$, $q = 23$, тогда модуль $n = p \cdot q = 17 \cdot 23 = 391$. Пусть $H_0 = 150$.

Тогда, используя (3.2), получим:

$$H_1 = (H_0 + m_1)^2 \bmod n = (150 + 17)^2 \bmod 391 = 27889 \bmod 391 = 128,$$

$$H_2 = (H_1 + m_2)^2 \bmod n = (128 + 16)^2 \bmod 39 = = 20736 \bmod 391 = 13,$$

$$H_3 = (H_2 + m_3)^2 \bmod n = (13 + 13)^2 \bmod 391 = = 676 \bmod 391 = 285,$$

$$H_4 = (H_3 + m_4)^2 \bmod n = (285 + 6)^2 \bmod 391 = = 84681 \bmod 391 = 225,$$

$$H_5 = (H_4 + m_5)^2 \bmod n = (225 + 19)^2 \bmod 391 = = 59536 \bmod 391 = 104,$$

$$H_6 = (H_5 + m_6)^2 \bmod n = (104 + 4)^2 \bmod 391 = = 11664 \bmod 391 = 325,$$

$$\mathrm{H}_7 = (\mathrm{H}_6 + \mathrm{m}_7)^2 \bmod n = (325 + 21)^2 \bmod 391 = = 119716 \bmod 391 = 70.$$

В результате хеш-образ сообщения «ПолесГУ» будет $h(M) = H_7 = 70$.

3.2. Электронная цифровая подпись

В 1976 г. Уитфилдом Диффи и Мартином Хеллманом было впервые предложено понятие «электронная цифровая подпись», хотя они предполагали, что схемы ЭЦП могут существовать.

Электронная цифровая подпись (ЭЦП) предназначена для защиты электронного документа, передаваемого посредством различных сред или хранящегося в цифровом виде, от подделки и является атрибутом электронного документа. Она получается в результате криптографического преобразования информации с использованием секретного ключа электронной цифровой подписи и позволяет идентифицировать владельца сертификата ключа подписи, установить отсутствие искажения информации в электронном документе.

Т.о., ЭЦП – это программно-криптографическое средство, которое обеспечивает:

- проверку целостности документов;
- конфиденциальность документов;
- установление лица, отправившего документ.

Цифровая подпись для электронных документов играет ту же роль, что и подпись, поставленная от руки в документах на бумаге. Но основное отличие ЭЦП от обычной подписи в том, что ЭЦП для каждого нового сообщения уникальна и не совпадает с подписями для других сообщений.

При разработке механизма цифровой подписи возникают следующие задачи:

1. Формирование подписи таким образом, чтобы ее невозможно было подделать.
2. Обеспечение возможности проверки того, что подпись действительно принадлежит указанному субъекту.
3. Предотвращение отказа субъекта от своей подписи.

Классическая схема создания цифровой подписи

До того как будет происходить формирование цифровой подписи, отправитель должен сгенерировать пару ключей: открытый K_O и секретный K_C.

При этом секретный ключ должен быть известен только отправителю (тому, кто подписывает сообщение), а открытый – любому желающему проверить подлинность сообщения. Часто подписывают не само сообщение, а его хеш-образ.

При создании цифровой подписи по классической схеме отправитель должен выполнить следующие действия:

1. Вычислить хеш-образ m исходного сообщения M при помощи некоторой хеш-функции h.
2. Вычислить цифровую подпись S по хеш-образу сообщения с использованием секретного ключа K_C.
3. Сформировать новое сообщение (M, S), состоящее из исходного сообщения и добавленной к нему цифровой подписи.

Классическая схема проверки цифровой подписи

Получив подписанное сообщение (M', S), получатель должен выполнить следующие действия для проверки подлинности подписи и целостности полученного сообщения:

1. Вычислить хеш-образ m' сообщения при помощи той же хеш-функции h.
2. С использованием открытого ключа (K_O) извлечь хеш-образ m сообщения из цифровой подписи S.
3. Сравнить вычисленное значение m' и m. Если хеш-образы совпадают, то подпись признается подлинной.

Фальсификация сообщения при его передаче по каналу связи возможна в случае, если злоумышленник узнает секретный ключ K_C или сможет провести успешную атаку против хеш-функции. Используемые в реальных приложениях хеш-функции обладают характеристиками, делающими атаку против цифровой подписи практически неосуществимой.

Например, хеш-функция SHA-1, принятая в США в качестве стандарта в 1995 г., формирует 160-битовый хеш-образ при обработке сообщения блоками по 512 бит.

С 2010 г. происходит переход от использования хеш-функции SHA-1 к хеш-функции SHA-2, которая может формировать хеш-образ длиной 224, 256, 512 или 1 024 бит.

3.3. Алгоритм цифровой подписи RSA

В 1977 г. в Массачусетском технологическом институте США Рональд Ривест, Ади Шамир и Леонард Адельман разработали криптографический алгоритм RSA, который без дополнительных модификаций можно использовать для создания примитивных цифровых подписей.

Алгоритм генерации ключей

Для формирования подписи по алгоритму RSA сначала необходимо вычислить пару ключей: открытый / секретный ключ, как это делается для криптосистемы RSA:

1. Выбираются два случайных простых числа p и q таких, что $p \approx q$.
2. Вычисляется их произведение $n = p \cdot q$.
3. Для значения n вычисляется функция Эйлера:

$$\varphi(n) = (p-1) \cdot (q-1).$$

4. Выбирается открытое число e такое, что:

$$1 < e < \varphi(n) \text{ и } \text{НОД}(e, \varphi(n)) = 1.$$

5. Вычисляется секретное число d такое, что выполняется следующее условие $(e \cdot d) \bmod \varphi(n) = 1$.

Открытый ключ $K_O = (e, n)$ автор передает партнерам по переписке для проверки его цифровых подписей.

Секретный ключ подписи $K_C = (d, n)$ сохраняется автором для подписи очередных сообщений.

Алгоритм постановки подписи

Чтобы подписать сообщение M, автор сообщения с помощью хеш-функции h вычисляет хеш-образ $m = h(M)$ исходного сообщения. Затем он вычисляет цифровую подпись S, используя хеш-образ m и секретное значение d по формуле:

$$S = m^d \bmod r. \qquad (3.3)$$

Пара $(M,\ S)$ передается получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа $(d,\ r)$.

Алгоритм проверки подписи

После приема пары (M', S) получатель вычисляет хеш-образ сообщения M' различными двумя способами.

Прежде всего, он восстанавливает хеш-образ m, применяя криптографическое преобразование подписи S с использованием открытого ключа (e, r):

$$m = S^e \bmod r. \qquad (3.4)$$

Кроме того, он находит результат хеширования m' принятого сообщения M' с помощью такой же хеш-функции h: $m' = h(M')$. Если вычисленные значения совпадают, т.е. $h(M) = S^e \bmod r$, то получатель признает пару (M', S) подлинной.

Пример 3.1. Подпишем сообщение «ПолесГУ» и выполним процедуру проверки подписи. Сначала получим его хеш-образ. Как показано выше, он равен $h(M) = 70$.

Далее сгенерируем открытый и закрытый ключи:

1. Выберем $p = 17$, $q = 23$.
2. Вычислим $r = 17 \cdot 23 = 391$.
3. Вычислим $\varphi(r) = (p-1) \cdot (q-1) = 16 \cdot 22 = 352$.
4. Выберем открытую экспоненту $e = 29$, взаимно простую с $\varphi(r) = 352$.
5. На основе e и $\varphi(r)$ вычислим число $d = 85$, используя расширенный алгоритм Евклида.

Тогда открытый ключ будет равен $(29{,}391)$, а закрытый – $(85{,}391)$. Далее подписываем сообщение $S = m^d \bmod r = = 70^{85} \bmod 391 = 185$, после чего отправляем сообщение, состоящее из самого текста и подписи: {ПолесГУ, 185}.

Допустим, что при передаче сообщение было изменено, и получателю доставлено сообщение {ПолемГУ, 185}. Для проверки подписи он сначала вычисляет хеш-образ полученного сообщения «ПолемГУ»:

$$H_1 = (H_0 + m_1)^2 \bmod n = (150 + 17)^2 \bmod 391 = = 27889 \bmod 391 = 128,$$

$$H_2 = (H_1 + m_2)^2 \bmod n = (128 + 16)^2 \bmod 391 =$$
$$= 20736 \bmod 391 = 13,$$

$$H_3 = (H_2 + m_3)^2 \bmod n = (13 + 13)^2 \bmod 391 =$$
$$= 676 \bmod 391 = 285,$$

$$H_4 = (H_3 + m_4)^2 \bmod n = (285 + 6)^2 \bmod 391 =$$
$$= 84681 \bmod 391 = 225,$$

$$H_5 = (H_4 + m_5)^2 \bmod n = (225 + 13)^2 \bmod 391 =$$
$$= 56644 \bmod 391 = 340,$$

$$H_6 = (H_5 + m_6)^2 \bmod n = (340 + 4)^2 \bmod 391 =$$
$$= 118336 \bmod 391 = 254,$$

$$H_7 = (H_6 + m_7)^2 \bmod n = (254 + 21)^2 \bmod 391 =$$
$$= 75625 \bmod 391 = 162.$$

С другой стороны, из цифровой подписи с помощью известного ему открытого ключа $(29, 391)$ получатель вычисляет хеш-образ, переданный отправителем:

$$S = m^e \bmod r = 185^{29} \bmod 391 = 70.$$

Т.к. два вычисленных значения 162 и 70 не равны, то подпись признается недействительной.

3.4. Алгоритм цифровой подписи DSA

Алгоритм DSA (Digital Signature Algorithm) был разработан в 1991 г. и с тех пор используется как стандарт США для электронной цифровой подписи – Digital Signature Standard (DSS).Согласно определению стандарта DSS, алгоритм DSA предусматривает применение в качестве хэш-функции алгоритма SHA. Заметим, что параметры алгоритма не засекречены. DSA основан на трудности вычисления дискретных логарифмов и базируется на схеме, первоначально представленной Т. Эль-Гамалем и К. Шнорром.

Алгоритм генерации ключей

Для получения пары «секретный / открытый» ключ необходимо выполнить следующие действия:

1. Выбрать большое простое число q.
2. Выбрать простое число p такое, что q является делителем числа $(p-1)$.
3. Подобрать число g такое, что для него верно:

$$g = h^{(p-1)/q} \bmod p,$$

где h – некоторое произвольное число из интервала $(1, p-1)$, и при этом $g > 1$. В большинстве случаев значение $h = 2$ удовлетворяет этому требованию.

4. Закрытый ключ отправителя x выбирается случайно из интервала $(0,\ q)$.
5. Открытый ключ вычисляется из закрытого ключа по формуле:

$$y = g^x \bmod p. \qquad (3.5)$$

Т.о., открытый ключ – $K_O = (p,\ q,\ y)$ (автор передает его партнерам по переписке для проверки его цифровых подписей), секретный – $K_C = x$ (служит для подписи автора сообщений). При этом значения p и q могут быть общими для группы пользователей, а значение y и x – для каждого свое.

Алгоритм постановки подписи

Подпись сообщения выполняют по следующему алгоритму:

1. Получаем хеш-образ исходного сообщения $h\,(M)$. При использовании формулы (3.2) вычисления необходимо выполнять по модулю числа q.
2. Выбирается случайное число $k \in (0, q)$, уникальное для каждой подписи.
3. Вычисляются значения r и s по формулам:

$$r = (g^k \bmod p) \bmod q, \qquad (3.6)$$

$$s = (k^{-1} \cdot (h(M) + x \cdot r)) \bmod q, \qquad (3.7)$$

где k^{-1} – это целое число, обратное k по модулю q, т.е. $(k^{-1} \cdot k) \bmod q = 1$ (k^{-1} вычисляется как в формуле (2.5)).

Если одно из полученных значений r или s будет равно 0, то необходимо повторить вычисления для другого значения k. В результате подписью будет пара значений $(r,\ s)$.

Т.о., сообщение с подписью будет иметь вид $\{M,\ r,\ s\}$.

Алгоритм проверки подписи

Для того чтобы проверить подлинность подписи, сначала из полученного сообщения $\{M',\ r,\ s\}$ вычисляется хеш-образ $h(M')$, после чего находят значение v, используя формулы (3.8).

Подпись признается подлинной, если $v = r$:

$$\begin{cases} w = s^{-1} \bmod q, \\ u_1 = (h(M') \cdot w\ ) \bmod q, \\ u_2 = (r \cdot w) \bmod q, \\ v = ((g^{u_1} \cdot y^{u_2}\ ) \bmod p) \bmod q, \end{cases} \quad (3.8)$$

где значение s^{-1} – целое число, которое можно получить, как в формуле (2.5).

Замечание. Недостаток алгоритма ЭЦП DSA – это наличие в нем ресурсоемких операций определения обратных величин по модулю q. Однако данный «минус» вполне компенсируется, если часть вычислений выполнить заранее.

Пример 3.2. Подпишем сообщение «ПолесГУ» с помощью алгоритма подписи DSA и выполним процедуру проверки. Как ранее было вычислено, хеш-образ сообщения «ПолесГУ» равен 70. Далее сгенерируем открытый и секретный ключи для создания подписи. Для этого выберем случайные простые числа q и p, пусть они будут равны соответственно 367 и 61. Как видно, $p - 1$ (366) делится на q (61) без остатка. Тогда число $g = 2^{(367-1)/61} \bmod 367 = 64$. Далее выберем случайное число $x = 35$, которое будет секретным ключом. Вычислим для него открытый ключ по формуле (3.5):

$$y = g^x \bmod p = 64^{35} \bmod 367 = 323.$$

Вычислим цифровую подпись для сообщения. Для этого возьмем его хеш-образ $h(M) = 70$, сгенерируем случайное число $k = 17$ и вычислим $r,\ s$ по формулам (3.6) – (3.7) ($k^{-1} = 18$ при $k = 17$, т.к. $(17 \cdot 18) \bmod 61 = 1$):

$$r = (g^k \bmod p) \bmod q = (64^{17} \bmod 367) \bmod 61 = \\ = 211 \bmod 61 = 28,$$

$$s = \left(k^{-1} \cdot (h(m) + x \cdot r)\right) \bmod q = \\ = (18 \cdot (70 + 35 \cdot 28)) \bmod 61 = 51.$$

Т.к. оба полученных значения r и s не равны 0, то подпись будет равна паре значений (28, 51), и отправляемое сообщение будет иметь вид: {ПолесГУ, 28,51}. Для проверки подлинности подписи получатель выполняет следующие действия. Сначала он вычисляет хеш-образ сообщения «ПолесГУ», которое равно 70. Далее вычисляет значение v по формулам (3.7):

$$\begin{cases} w = s^{-1} \bmod q = 6 \bmod 61 = 6, \\ u_1 = (h(M) \cdot w) \bmod q = (70 \cdot 6) \bmod 61 = 54, \\ u_2 = (r \cdot w) \bmod q = (28 \cdot 6) \bmod 61 = 46, \\ v = (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q = \\ = ((64^{54} \cdot 323^{46}) \bmod 367) \bmod 61 = 28. \end{cases} \quad (3.8)$$

Т.к. $r = v$ $(28 = 28)$, то подпись является подлинной.

Задание для выполнения лабораторной работы № 3

1. Изучить теоретический материал по лабораторной работе.

2. Реализовать программу вычисления и проверки электронной цифровой подписи согласно варианту в **Таблице 3.1**.

Для вычисления хеш-образа сообщения использовать функцию (3.2).

Таблица 3.1. – Электронная цифровая подпись

Вариант	Схема
№ 1	DSA
№ 2	RSA

3. Сформировать отчет о проделанной работе.

4. ДОКАЗАТЕЛЬСТВО С НУЛЕВЫМ РАЗГЛАШЕНИЕМ ЗНАНИЯ

Одна из основных задач криптографии представляет собой двустороннюю интерактивную игру, в которой один участник (доказывающая сторона) доказывает другому участнику (проверяющей стороне) истинность утверждения, не раскрывая сущности доказательства.

Представим себе, что утверждение, которое необходимо доказать, не раскрывая сущности доказательства, является решением какой-либо знаменитой нерешенной математической задачи. В этом случае доказывающая сторона, опасающаяся плагиата, может пожелать скрыть технические детали доказательства от потенциально нечестного рецензента. Для этого она должна провести «секретное» доказательство, убедив рецензента (играющего роль проверяющей стороны) в корректности выводов, не давая никакой дополнительной информации. Рассматривая такие доказательства, необходимо изучить два вопроса:

Вопрос 1. Сколько информации получит проверяющая сторона в ходе интерактивного доказательства?

Вопрос 2. Сколько раундов должна выполнить доказывающая сторона, чтобы убедить проверяющего?

Идеальным ответом на первый вопрос был бы «нисколько». IP-протокол (IP – interactive proof), обладающий таким свойством, называется протоколом с нулевым разглашением или ZK-протоколом (zero-knowledge – ZK).

Второй вопрос важен не только для практических приложений, но и для теории вычислительной сложности, поскольку решение этой проблемы связано с получением более низкой оценки сложности.

Доказательство нулевым разглашением был придумано и разработано в 1985 г. учеными Шафи Гольдвассером, Сильвио Микалием и Чарльз Реккофом.

Т.о., доказательство с нулевым разглашением – это интерактивный протокол, позволяющий одной из сторон (проверя-

ющему) убедиться в достоверности какого-либо утверждения (обычно математического), не получив при этом никакой другой информации от второй стороны (доказывающего).

Доказательство с нулевым разглашением знания должно обладать тремя свойствами:

1. Полнота: если утверждение действительно верно, то доказывающий убедит в этом проверяющего.

2. Корректность: если утверждение неверно, то даже нечестный доказывающий не сможет убедить проверяющего, за исключением пренебрежимо малой вероятности.

3. Нулевое разглашение: если утверждение верно, то любой даже нечестный проверяющий не узнает ничего, кроме самого факта, что утверждение верно.

Обозначим основную модель протокола интерактивных доказательств через (P, V), где P – доказывающая сторона (prover), V – проверяющая (verifier).

4.1. Алгоритм Фиата-Шамира

Одним из наиболее известных протоколов идентификации личности с помощью доказательства с нулевым знанием является протокол, предложенный Амосом Фиатом и Ади Шамиром.

Стойкость данного протокола основывается на сложности извлечения квадратного корня по модулю достаточно большого составного числа n, факторизация которого неизвестна. Доверенный центр T выбирает и публикует число $n = p \cdot q$, где $p > q$ – простые числа и держатся в секрете, при этом n достаточно большое число, разложить на множители которое трудно. Каждый пользователь P выбирает секретное $s \epsilon [1,\ n-1]$ взаимно простое с n. Затем вычисляется открытый ключ:

$$v = s^2 \bmod n. \qquad (4.1)$$

Полученное v регистрируется доверенным центром в качестве открытого ключа пользователя V, а значение s является секретом V.

Именно принадлежность секретного ключа s необходимо доказать P некоторой стороне V без разглашения секрета s за t раундов (аккредитаций).

Каждая аккредитация состоит из следующих этапов:

1. $P \to V$ (доказывающий отсылает проверяющему) число $x = r^2 \bmod n$, где случайное число $r \epsilon [1, n-1]$.
2. $V \to P$ случайно выбранный бит $e \epsilon \{0, 1\}$ (т.е. 0 или 1).
3. $P \to V$ число $y = (r \cdot s^e) \bmod n$.
4. V проверяет равенство $y^2 \equiv (x \cdot v^e) \bmod n$. Если оно верно, то происходит переход к следующему раунду протокола, иначе раунд не пройден, и доказательство прекращается.

Важное замечание. Вероятность того, что пользователь P не знал секрета s, но убедил в обратном проверяющего V, оценивается вероятностью, равной $p = 2^{-t}$, где t – число раундов. Для достижения высокой достоверности число раундов выбирают достаточно большим ($t \in [20; 40]$).

Т.о., V удостоверяется в знании P секрета тогда и только тогда, когда все t раундов прошли успешно.

Пример 4.1. Рассмотрим работу алгоритма Фиата – Шамира на маленьких числах. Пусть доверенный центр выбрал простые $p = 11$ и $q = 13$, тогда:

$$n = 11 \cdot 13 = 143.$$

Пользователь P выбирает секрет $s = 41$, откуда по формуле (4.1) вычисляет открытый ключ:

$$\mathrm{v} = 41^2 \bmod 143 = 1681 \bmod 143 = 108.$$

Т.о., в секрете держатся 11, 13 и 41, а 108 и 143 публикуются открыто.

Рассмотрим один из раундов доказательства стороне V, что открытый ключ v принадлежит именно P тем, что P знает секрет s:

1. P выбирает $r = 15$ и считает:

$$x = 15^2 \bmod 143 = 225 \bmod 143 = 82.$$

2. V выбирает бит $e \in \{0, 1\}$ и отсылает его P.

3. Если V отправил $e = 0$, то P возвращает $y = r \bmod n = 15$, иначе P возвращает:

$$y = (r \cdot s)(\bmod n) = (15 \cdot 41) \bmod 143 = 615 \bmod 143 = 43.$$

4. V проверяет полученные значения.

Если $e = 0$, то $y^2 \bmod n = 15^2 \bmod 143 = 82$ и $x = 82$, что подтверждает знание секрета s стороной P. Иначе, если $e = 1$, то $y^2 = 43^2 \bmod 143 = 133$ и $(x \cdot v) \bmod 143 = (82 \cdot 108) \bmod 143 = 133$, что также подтверждает знание секрета s стороной P.

Замечание. Для того чтобы этот протокол корректно выполнялся, P никогда не должен повторно использовать значение x. Иначе V во время другого раунда отправил бы P на шаге 2 другой случайный бит e и имел бы оба ответа P. После этого V мог бы вычислить значение s, и ему стал бы известен секрет P.

4.2. Алгоритм Гиллу-Кискатра

Алгоритм Гиллу-Кискатра является развитием схемы Фиата-Шамира.

Он позволяет сократить число аккредитаций t (часто до одной) и объем используемой памяти. Согласно ему доверенный центр выбирает и публикует модуль $n = p \cdot q$, где p, q – простые секретные числа, найти которые, зная n, невозможно, и открытое число $v \geq 3$ такое, что $\text{НОД}\big(v, \varphi(n)\big) = 1$ ($\varphi(n) = (p - 1)(q - 1)$).

Используя эти данные, вычисляется секретное число:

$$s = v^{-1} \bmod \varphi(n). \qquad (4.3)$$

Параметры (v, n) являются общедоступными и могут быть использованы для генерации открытого и закрытого ключей группой пользователей. Далее каждому пользователю доверенным центром выдается некоторый публичный

идентификатор его личности Id, при этом $1 < Id < n$, и вычисляется некоторый секрет s_P:

$$s_P = Id^{-s} \bmod n. \quad (4.4)$$

Сторона P посылает проверяющему V свои атрибуты Id. И для доказательства того, что Id – это именно его идентификатор, пользователь P должен убедить V, что ему известен секрет s_P:

1. $P \to V$ $x = r^v \bmod n$, где $r \in [1, n-1]$ – случайное целое число.
2. $V \to P$ случайное целое $e \in [\,1, v-1]$.
3. $P \to V$ значение:

$$y = r{s_P}^e \bmod n. \quad (4.5)$$

4. V проверяет равенство:

$$x = y^v Id^e \bmod n, \quad (4.6)$$

и если оно верно, то раунд пройден успешно.

Пример 4.2. Рассмотрим работу указанного протокола на маленьких числах. Для этого возьмем $n = p \cdot q = 11 \cdot 13 = 143$ и $v = 37$, тогда:

$$\varphi(n) = \varphi(143) = (11-1)(13-1) = 120.$$

Вычислим s, согласно формуле (4.3) (см. также (2.1)):

$$s = v^{-1} \bmod \varphi(n) = 37^{-1} \bmod 120 = 37^{\varphi(\varphi(n))-1} \bmod 120 = \\ = 37^{31} \bmod 120 = 13.$$

Далее присвоим пользователю P идентификатор $Id = 29$ и вычислим его секрет s_P по формуле (4.4):

$$s_P = Id^{-s} \bmod n = 29^{-13} \bmod 143 = 29^{13\cdot(\varphi(143)-1)} \bmod 143 = \\ = 29^{13\cdot 119} \bmod 143 = 61.$$

Для доказательства знания секрета s_P без его разглашения пользователь P действует согласно схеме:

1. $P \to V$ число $x = 21^{37} \bmod 143 = 21$ ($r = 21$ – случайное число).
2. $V \to P$ случайное целое $e = 17$.
3. $P \to V$ число $y = (rs_P{}^e) \bmod n = (21 \cdot 61^{17}) \bmod 143 = 102$.
4. Т.к. $(y^v Id^e) \bmod n = (102^{37} 29^{17}) \bmod 143 = 21$, то раунд пройден успешно.

4.3. Алгоритм Шнорра

Особенностью алгоритма Шнорра является то, что он позволяет проводить предварительные вычисления, что удобно при малых вычислительных ресурсах. Надежность данного алгоритма основывается на сложности вычисления дискретного логарифма.

Доверенный центр выбирает два простых числа p и q таких, что $p - 1$ делится без остатка на q, и выбирается элемент $g \neq 1$ такой, что для него верно:

$$g = h^{(p-1)/q} \bmod p,$$

где h – некоторое произвольное число из интервала $(1, p - 1)$.

Параметры (p, q, g) являются открытой информацией и свободно публикуются. Они могут быть общими для группы пользователей. Также выбирается параметр t $(40 \leq t < q)$, определяющий уровень безопасности. Далее доказывающая сторона P выбирает секрет $[1, q - 1]$ и вычисляет:

$$v = g^{-s} \bmod p, \qquad (4.7)$$

где v является открытым ключом P, который посылается доверенному центру и там публикуется как открытое значение пользователя P.

Для доказательства знания секрета s проверяющему V пользователь P выполняет следующие действия:

1. $P \to Vx = g^r \bmod p$, где случайное число $r \in [1, q - 1]$.
2. $V \to P$ случайное число e, где $e \in (1; 2^t - 1)$.
3. $P \to V$ число $y = (s \cdot e + r) \bmod q$.

4. V проверяет равенство: $z = g^y \cdot v^e \bmod p$. (4.8)

Если равенство верно, то раунд пройден успешно.

Пример 4.3. Приведем работу указанного алгоритма. Пусть $p = 643$ и $q = 107$, при этом выполняется условие, что $107|642$ (читается – «107 делит число 642»).

Далее вычислим g из условия $g = h^{\frac{p-1}{q}} \bmod p$, где h – некоторое произвольное число из интервала $(1, p-1)$. В данном случае для $h = 2$ получим $g = 64$. Сторона P выбирает закрытый ключ $s = 43$ и вычисляет по формуле (4.7) открытый ключ:

$$v = g^{-s} \pmod p = 64^{-43} \bmod 643 = 623.$$

Доказательство происходит следующим образом:

1. $P \to V$ число $x = g^r \bmod p = 64^{26} \bmod 643 = 516$, где случайное число $r = 26$.
2. $V \to P$ число $e = 19$.
3. $P \to V$ число $y = (s \cdot e + r) \bmod q = (43 \cdot 19 + 26) \bmod 107 = 94$.
4. V вычисляет значение:

$(g^y \cdot v^e) \bmod p = (64^{94} \cdot 623^{19}) \bmod 643 = 516$, что равно x.

Задание для выполнения лабораторной работы № 4

1. Изучить теоретический материал по лабораторной работе.
2. Реализовать алгоритм доказательства с нулевым разглашением знания по алгоритму согласно своему варианту **Таблицы 4.1**.
3. Сформировать отчет о проделанной работе.

Таблица 4.1 – Доказательство с нулевым разглашением

Вариант	**№ 1**	**№ 2**	**№ 3**
Алгоритм	Фиата-Шамира	Гиллу-Кискатра	Шнорра

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бубнов, С.А. Лабораторный практикум по основам криптографии : учебно-методическое пособие для студентов по профилю подготовки 080801.65 «Прикладная информатика (в экономике)» / С.А. Бубнов. – Саратов, 2012. – 35 с.

2. Венбо, М. Современная криптография: теория и практика / М. Венбо. – М. : Изд. дом «Вильямс», 2005. – 768 с.

3. Виноградов, И.М. Основы теории чисел / И.М. Виноградов. – М. : Наука, 1981. – 176 с.

4. Гинзбург, А.И. Пластиковые карты / А.И. Гинзбург. – СПб. : Питер, 2004. – 128 с.

5. Грибунин, В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. – М. : СОЛОН-Пресс. 2002. – 272 с.

6. Деднев, М.А. Защита информации в банковском деле и электронном бизнесе / М.А. Деднев, Д.В. Дыльнов. – М. : Кудиц-Образ, 2004. – 512 с.

7. Кнут, Д. Искусство программирования для ЭВМ : в 3-х т. / Д. Кнут. – М. : Мир, 1977. – Т. 2 : Получисленные методы. – 724 с.

8. Лидл, Р. Конечные поля / Р. Лидл, Т. Нидеррайтер. – М. : Мир, 1988. – 820 с.

9. Харин, Ю.С. Математические и компьютерные основы криптологии / Ю.С. Харин [и др.]. – Минск : Новое знание, 2003. – 382 с.

10. Молдовян, Н.А. Введение в криптосистемы с открытым ключом: Проблематика криптографии; элементы теории чисел; двухключевые криптосистемы и др. : учеб. пособие для вузов / Н.А. Молдовян, А.А. Молдовян. – СПб. : БХВ-Петербург, 2005. – 288 с.

11. Романец, Ю.В. Защита информации в компьютерных системах и сетях / Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин. – М. : Радио и связь, 1999. – 328 с.

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

13. Харин, Ю.С. Компьютерный практикум по математическим методам защиты информации / Ю.С. Харин, С.В. Агиевич. – Минск : БГУ, 2001. – 190 с.

14. Харин, Ю.С. Математические основы криптологии / Ю.С. Харин, В.И. Берник, Г.В. Матвеев. – Минск : БГУ, 1999. – 319 с.

15. Шеннон, К. Работы по теории информации и кибернетике / К. Шеннон. – М. : Издательство иностранной литературы, 1963. – 830 с.

16. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнаейр. – М. : ТРИУМФ, 2002. – 816 с.

17. Ярмолик, В.Н. Криптография, стеганография и охрана авторского права / В.Н. Ярмолик, С.С. Портянко, С.В. Ярмолик. – Минск : Изд. центр БГУ, 2007. – 240 с.

18. Ярмолик, В.Н. Элементы теории информации : практикум для студ. спец. «Программное обеспечение информационных технологий» / В.Н. Ярмолик, П. Занкович, С.С. Портянко. – Минск : БГУИР, 2007. – 40 с.

19. Ярмолик, С.В. Криптография и охрана коммерческой информации. Методическое пособие по выполнению лабораторных работ для студентов специальностей 1-40 01 02-02 и 1-26 02 03 дневной и заочной форм обучения / С.В. Ярмолик, В.Н. Ярмолик. – Минск : БГУИР, 2011. – 33 с.

Учебное издание

Романова Марина Александровна

Криптография и защита информации

Учебно-методическое пособие

Ответственный за выпуск *П.Б. Пигаль*

Подписано в печать 11.03.2016 г. Формат 60×84/16.
Бумага офсетная. Гарнитура «Таймс». Ризография.
Усл. печ. л. 2,73. Уч.-изд. 1,43 л.
Тираж 80 экз. Заказ № 157.

Отпечатано в редакционно-издательском отделе
Полесского государственного университета
225710, г. Пинск, ул. Днепровской флотилии, 23.