Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный технический университет имени
П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

И.А. Мурашко

ЗАЩИТА КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ

Лабораторный практикум по одноименной дисциплине
для студентов направления специальности
1–40 01 02 -01 – Информационные системы и технологии
(в проектировании и производстве)
дневной и заочной формы обучения

Гомель 2014

Лабораторная работа №1
Простейшие алгоритмы шифрования
(2+2 часа)

Цель работы: Изучить простейшие алгоритмы шифрования.

1 ПЕРЕСТАНОВОЧНЫЕ ШИФРЫ

1.1 Простейшие перестановочные шифры

С древних времен для сокрытия смысла записанного сообщения люди использовали различные хитрости. Например, удаляли пробелы и писали слова только большими (только малыми) буквами.

Это лекция по алгоритмам→ ЭТОЛЕКЦИЯПОАЛГОРИТМАМ

Следующим шагом усложнения является разбиение зашифрованного текста на блоки.

ЭТОЛ ЕКЦИ ЯПОА ЛГОР ИТМА М

Эффективным способом шифрования является запись слов в обратном порядке.

ОТЭ ЯИЦКЕЛ ОП МАМТИРОГЛА

В общем случае перестановочный шифр переставляет символы исходного текста по определенной схеме.

Перестановка может быть представлена в виде геометрической фигуры.

Исходный текст *M*	==========> Правило записи	**Фигура**	==========> Правило чтения	Зашифрованный текст *C*

M –, *C* –

Пример: матрица 2 строки, 5 столбцов. Запись построчная. Чтение по столбцам сверху вниз 4, 1, 2, 5, 3.

M: шифрование

	1	2	3	4	5
1	**ш**	**и**	**ф**	**р**	**о**
2	**в**	**а**	**н**	**и**	**е**

C: ришвиаоефн

Формально эта процедура записывается следующим образом: исходный текст *M* разбивается на блоки $M = m_1, m_2, \ldots m_i$, все блоки одинаковой длинны. Тогда зашифрованный текст будет представлен как совокупность блоков исходного текста преобразованных в соответствии с функцией f.

$$E_k(M) = m_{f1},m_{f2},\ldots,m_{fi}$$

Дешифрация выполняется обратным образом.

Все символы исходного текста появятся в зашифрованном тексте.

1.2 Шифр типа «Железнодорожная изгородь»

Пусть имеется правило записи текста следующего вида:

1 5 9 13
2 4 6 8 10 12 14
3 7 11 15

Геометрическая фигура соответствует изгороди. В этом случае исходный текст «ЭТОЛЕКЦИЯПОШИФРАМ» будет записан следующим образом:

Э Е Я И М
Т Л К И П Ш Ф А
О Ц О Р

При использовании правила чтения по строкам слева направо начиная с первой строки будет получен следующий шифротекст:
«ЭЕЯИМТЛКИПШФАОЦОР».

1.3 Ключевое слово или ключевая фраза

Одной из наиболее известных модификаций метода перестановки является использование ключевого слова или фразы в качестве правила перестановки столбцов.

Пример: КРИПТОГРАФИЯ может быть использовано, как ключ. Буквам ключевого слова назначаются номера, начиная с первого, в соответствии с алфавитом. Если буква встречается несколько раз, то нумерация определяется порядком следования повторяющейся буквы в ключевом слове (запись построчно, чтение по столбцам, начиная с первого столбца).

К 5	**Р** 8	**И** 3	**П** 7	**Т** 10	**О** 6	**Г** 2	**Р** 9	**А** 1	**Ф** 11	**И** 4	**Я** 12
Э Л	Т Г	О О	Л Р	Е И	К Т	Ц М	И А	Я М	П Ш	О И	А Ф
Р	О	В	А	Н	И	Я					

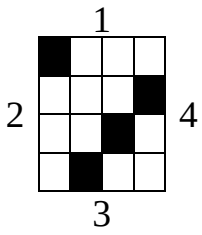
В результате будет получен следующий шифротекст «ЯМЦМЯООВОИЭЛРКТИЛРАТГОИАЕИНПШАФ».

1.4 Метод поворачивающейся решетки

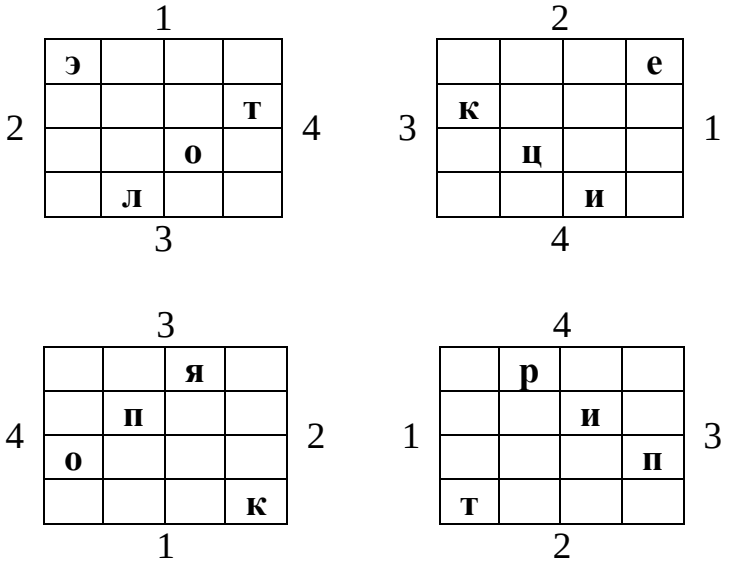
Суть метода: исходный текст записывается через отверстия в решетке, которая по мере заполнения поворачивается на 90º. Предварительно текст разбивается на блоки (в данном случае блок равен 16 символам).

Пример.

■ - вырезанные ячейки (куда вписываем текст)



Исходный текст: «ЭТОЛЕКЦИЯПОКРИПТ|ОГРАФИИАБВ…», запись – построчная



э	р	я	е
к	п	и	т
о	ц	о	п
т	л	и	к

=>

о	е	б	ф
и	в	ё	г
г	и	р	ж
з	а	а	а

первые 16 символов — остальные 16 символов

Криптотекст: «ЭРЯЕКПИТОИОПТЛИК|ОЕБФИВЁГГИ…».

Как изготовить решетку:

– строится матрица (4х4);

– ячейки матрицы, которые при повороте матрицы на 90о занимают одинаковое положение нумеруются одинаково;

– вырезается один из квадратов с одинаковым номером.

Например:

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

2 ПОДСТАНОВОЧНЫЕ ШРИФТЫ

2.1 Простейшие подстановочные шрифты

Изыскания в части создания эффективных подстановочных шифраторов были направлены в сторону поиска математического описания процедуры шифрования. В этом случае таблица подстановки присутствует в неявном виде, что существенно упрощает как саму процедуру шифрования, так и использование подобных систем на практике. Классическим примером подобных криптосистем является шифр Цезаря. Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке с Цицероном.

Таблица подстановки присутствует в неявном виде, т.е. символ шифротекста вычисляется по математическому выражению:

$$c_i = (a_i + k) \; mod \; n,$$

где a_i – символ исходного текста;

k – ключ;

n – мощность алфавита.

В шифре Цезаря используется $k = 3$.

Для английского алфавита, используя последовательную нумерацию

букв 0–*A*, 1–*B*, 2–*C*, 3–*D*, 4–*E*, 5–*F*, 6–*G*, 7–*H*, 8–*I*, 9–*J*, 10–*K*, 11–*L*, 12–*M*, 13–*N*, 14–*O*, 15–*P*, 16–*Q*, 17–*R*, 18–*S*, 19–*T*, 20–*U*, 21–*V*, 22–*W*, 23–*X*, 24–*Y*, 25–*Z*, процедура шифрования, предложенная Цезарем, будет описываться соотношением

$$c_i = (a_i + 3) \ mod \ 26.$$

В данном случае и a_i, и c_i представляют собой номера букв в исходном алфавите. При шифровании буквы *B* исходного алфавита имеющей номер 6 получим c_i =1+3=4 что соответствует букве *E* используемой в качестве подстановочного элемента в шифротексте.

Пример. При шифровании исходного текста *M*=C*R*Y*P*TOG*R*A*P*HY получим *C*=FUBSW*R*JUDSKB.

Развитием этого метода является метод, основанный на свойстве децимации (децимация – выборка *k*-тых элементов):

$$c_i = (a_i * k) \ mod \ n.$$

То есть номера символов в шифротексте в *k* раз больше номеров символов исходного текста, например

0 1 2 3

A B C D

0 3 6 9

A D F G

Аффинное преобразование:

$$c_i = (k_1 a_i + k_2) \ mod \ n.$$

В данном случае используется 2 ключа: k_1 и k_2. Причем накладывается требование взаимной простоты $(k_1, n) = 1$.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Разработать программу, выполняющую шифрование текста не менее чем двумя перестановочными методами.

3.2 Разработать программу, выполняющую шифрование текста подстановочным методом (аффинное преобразование).

3.3 Сохранить полученный криптотекст в текстовый файл.

3.4 Разработать программу, выполняющую расшифровку криптотекста выбранными перестановочными методами.

3.5 Разработать программу, выполняющую расшифровку криптотекста подстановочным методом (аффинное преобразование).

3.6 Оформить отчет по проделанной работе.

В качестве исходного текста взять Фамилию_Имя,Отчество (не менее 20 символов).

Лабораторная работа №2
Математические основы криптографии
(4 часа)

Цель работы: Изучить основные математические преобразования, используемые в криптографии.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Наибольший общий делитель

Наибольшим общим делителем (НОД, или англ. – *Greatest Common Divider – GCD*) целых чисел $a_1, a_2, \ldots, a_n$ называется такой положительный общий делитель этих чисел, который делится на любой другой общий делитель этих чисел.

Пример:
НОД(21,15)=3;
НОД(27,44)=1;
НОД(120,66)=6.

1.2 Алгоритм Евклида

Используется для нахождения наибольшего общего делителя двух чисел. Идея:

НОД(a, b) = НОД(b, r), где

$$a = b \cdot q + r.$$

Пример: НОД(22, 8) = ?
22 = 8*2 + 6
(22, 8) = (8, 6)
8 = 6*1 + 2
(8, 6) = (6, 2)
6 = 2*2 + 2
(6, 2) = (2, 2)
2 = 2*1 + 0
Получили: НОД(22, 8) = 2.

1.3 Бинарный алгоритм

Данный алгоритм также используется для нахождения наибольшего общего делителя 2-х чисел и базируется на следующих четырёх утверждениях:

1) если оба числа a и b – четные, то: НОД(a, b) = 2·НОД(a/2, b/2);
2) если a – четное, b – нечетное, то НОД(a, b) = НОД(a/2, b);
3) НОД(a, b) = НОД(b, a– b);
4) если a и b –нечетны, то (a – b) – четно.

Пример: НОД(1173, 323) = ?

(1173, 323) = (323, 850) = (323, 425) = (323, 102) = (323, 51) = (51, 272) = (51, 136) = (51, 68) = (51, 34) = (51, 17) = (17, 34) = (17, 17) = 17
Получили: НОД(1173, 323) = 17.

1.4 Простые числа

Положительное целое не равное нулю число называется простым, если оно делится только на самого себя и на единицу.

Примеры: 11 – простое; 29 – простое; 56 – составное (56 = 7 · 4 · 2).

Два числа m и n называются взаимно простыми, если они не имеют общих делителей кроме единицы, то есть наибольший общий делитель НОД(m, n)=1.

1.5 Функция Эйлера

Функцией Эйлера φ(n) ($n \geq 1$) называют число положительных целых чисел меньших n и взаимно простых с n.

Примеры: φ(1) = 0; φ(2) = 1; φ(3) = 2; φ(4) = 2; φ(5) = 4; φ(6) = 2; φ(7) = 6; φ(8) = 4; φ(9) = 6; φ(10) = 4; φ(11) = 10.

Если n – простое число, то φ(n)=n – 1.

Пример: φ(31) =30.

Если n=p·q, где p и q – простые числа, то φ(n)=(p–1)·(q–1).

Пример: φ(35) = φ(5) · φ(7) = 4 · 6 = 24.

Обобщенный алгоритм вычисления функции Эйлера для произвольного числа n.

Если n представить как произведение простых чисел в соответствующих степенях:

$n = p_1^{a_1} \cdot p_2^{a_2} \cdot p_3^{a_3} \cdot \ldots \cdot p_r^{a_r}$ (p_1, p_2, …, p_r – простые),

то

$$\varphi(n)=n\cdot(1-1/p_1)\cdot(1-1/p_2)\cdot\ldots\cdot(1-1/p_r).$$

Пример:

φ (2700) = ?
2700 можно представить как $2^2 \cdot 3^3 \cdot 5^2$.
Тогда φ(2700)=2700·(1–1/2)·(1–1/3)·(1–1/5)=720.

1.6 Теорема Эйлера

Если $n \geq 0$ – положительное целое число и (a, n)=1, где a – целое, то справедливо:

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

Пример: 41^{126} = ? mod 127.
НОД(41,127) = 1, φ (127) = 126, поэтому 41^{126} = 1 mod 127.
Пример: 4^{336} = ? mod 377.
НОД(336,377) = 1, 377 = 13·29, 13 и 29 – простые числа, поэтому
φ(377) = φ(13) · φ(29) = 12·28 = 336, поэтому 4^{336} = 1 mod 377.

1.7 Взаимообратные числа

Для числа n взаимообратным по модулю r называется такое число m, для которого выполняется:

$$(n \cdot m) \bmod r = 1,$$

или

$$n \cdot m = 1 \bmod r. \qquad (1)$$

Доказательство: по теореме Эйлера: $n^{\varphi(n)}=1 \bmod r$ или

$$1 = n^{\varphi(n)} \bmod r. \qquad (2)$$

Перемножив (1) и (2), получим: $n \cdot m = n^{\varphi(n)} \bmod r$, разделив обе части на n, получим: $m = n^{\varphi(n)-1} \bmod r$.
Пример:
Найти взаимообратное по модулю 7 для числа 4.
$4 \cdot m = 1 \bmod 7$, $m = 4^{\varphi(7)-1} \bmod 7 = 4^5 \bmod 7 = 2$.
Проверка: 4·2 mod 7 = 8 mod 7 =1.

1.8 Принципы модулярной арифметики

Модулярная арифметика основывается на следующем равенстве:

$$(a*b) \bmod m = [(a \bmod m)*(b \bmod m)] \bmod m, \qquad (3)$$

где * – любая из следующих операций: “+” (сложение); “–” (вычитание); “× ” (умножение).

Данное равенство говорит о том, что вычисление $(a*b) \bmod m$ в модулярной арифметике даёт тот же результат что и вычисление $(a*b)$ в обычной целочисленной арифметике с последующим взятием остатка от деления полученного результата на m (mod m).

Пример: $7 \cdot 9 \bmod 5 = [(7 \bmod 5) \cdot (9 \bmod 5)] \bmod 5$.

Принципы модулярной арифметики также применимы к операции возведения в степень, поскольку возведение в степень эквивалентно многократному умножению.

Пример: Рассмотрим выражение $3^5 \bmod 7$. Возведение 3 в степень 5 и затем взятие результата по модулю 7 может быть произведено следующим образом:

Поскольку $5 = 2 \cdot 2 + 1$, то $3^5 = 3^{2 \cdot 2 + 1} = (3^2)^2 \cdot 3^1$.

1) Возводим число 3 в квадрат: $3 \cdot 3 = 9$ (3^2)
2) Возводим результат в квадрат: $9 \cdot 9 = 81$ ($(3^2)^2$)
3) Умножаем на 3: $81 \cdot 3 = 243$ ($(3^2)^2 \cdot 3$)
4) Берём по модулю 7: $243 \bmod 7 = 5$.

Каждый из полученных промежуточных результатов может быть взят по модулю 7.

1) Возводим число 3 в квадрат: $3 \cdot 3 \bmod 7 = 2$
2) Возводим результат в квадрат: $2 \cdot 2 \bmod 7 = 4$
3) Умножаем на 3: $4 \cdot 3 \bmod 7 = 5$.

2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

2.1 Разработать программу, выполняющую шифрование текста не менее чем двумя перестановочными методами.

2.2 Разработать программу, выполняющую шифрование текста подстановочным методом (аффинное преобразование).

Задание по лабораторной работе №2

1. Вычислите НОД (m, n) по алгоритму Евклида.
2. Вычислите НОД(m, n) используя бинарный алгоритм.
3. Вычислите функцию Эйлера для n=… (произвольное число)
4. Покажите, что $a^b = 1 \bmod n$
5. Вычислите взаимообратное число для m по модулю r.
6. Покажите, что $m^n \bmod r = k$.

Вариант 1

1. m=347, n=723;
2. m=112, n=679;
3. n=324;
4. a=4, b=336, n=377;
5. r=29, m=5;
6. m=8, n=7, r=13, k=5.

Вариант 2

1. m=552, n=874;
2. m=1934, n=725;
3. n=863;
4. a=6, b=480, n=527;
5. r=23, m=3;
6. m=5, n=8, r=19, k=4.

Вариант 3

1. m=1236, n=935;
2. m=1778, n=994;
3. n=632;
4. a=10, b=220, n=253;
5. r=26, m=5;
6. m=7, n=9, r=17, k=10.

Вариант 4

1. m=845, n=652;
2. m=964, n=1277;
3. n=953;
4. a=14, b=448, n=493;
5. r=55, m=6;
6. m=9, n=8, r=13, k=3.

Вариант 5

1. m=1974, n=528;
2. m=998, n=1285;
3. n=746;
4. a=13, b=504, n=551;
5. r=91, m=8;
6. m=9, n=7, r=19, k=4.

Вариант 6

1. m=1532, n=643;
2. m=994, n=778;
3. n=488;
4. a=13, b=672, n=731;
5. r=13, m=5;
6. m=4, n=11, r=7, k=13.

Вариант 7

1. m=2674, n=1699;
2. m=2674, n=1118;
3. n=886;
4. a=7, b=264, n=299;
5. r=18, m=7;
6. m=9, n=7, r=17, k=2.

Вариант 8

1. m=933, n=525;
2. m=525, n=1385;
3. n=724;
4. a=12, b=648, n=703;
5. r=26, m=11;
6. m=6, n=12, r=13, k=1.

Вариант 9

1. m=835, n=1562;
2. m=838, n=1200;
3. n=678;
4. a=8, b=360, n=407;
5. r=33, m=5;
6. m=8, n=9, r=19, k=18.

Вариант 10

1. m=2525, n=1186;
2. m=745, n=1375;
3. n=884;
4. a=15, b=756, n=817;
5. r=44, m=7;
6. m=5, n=9, r=17, k=12.

Вариант 11

1. m=472, n=844;
2. m=844, n=1483;
3. n=625;
4. a=24, b=936, n=1007;
5. r=28, m=7;
6. m=5, n=8, r=19, k=4.

Вариант 12

1. m=552, n=938;
2. m=938, n=1366;
3. n=728;
4. a=9, b=832, n=901;
5. r=32, m=9;
6. m=6, n=9, r=23, k=16.

Вариант 13

1. m=702, n=1157;
2. m=774, n=1266;
3. n=872;
4. a=8, b=624, n=689;
5. r=18, m=5;
6. m=8, n=9, r=17, k=8.

Вариант 14

1. m=1045, n=836;
2. m=686, n=1078;
3. n=842;
4. a=5, b=520, n=583;
5. r=25, m=6;
6. m=5, n=7, r=11, k=3.

Вариант 15

1. m=1265, n=2024;
2. m=1092, n=689;
3. n=634;
4. a=6, b=312, n=371;
5. r=16, m=3;
6. m=5, n=9, r=11, k=9.

Вариант 16

1. m=686, n=1078;
2. m=1045, n=836;
3. n=556;
4. a=8, b=624, n=689;
5. r=32, m=9;
6. m=8, n=9, r=19, k=18.

Вариант 17

1. m=1092, n=689;
2. m=1265, n=2024;
3. n=734;
4. a=14, b=448, n=493;
5. r=29, m=5;
6. m=5, n=8, r=19, k=4.

Вариант 18

1. m=938, n=1366;
2. m=552, n=938;
3. n=867;
4. a=7, b=264, n=299;
5. r=91, m=8;
6. m=9, n=8, r=13, k=3.

Вариант 19

1. m=994, n=778;
2. m=1532, n=643;
3. n=462;
4. a=10, b=220, n=253;
5. r=44, m=7;
6. m=4, n=11,r=17, k=13

1. Вариант 20
2. m=112, n=679;
3. m=347, n=723;
4. n=532;
5. a=15, b=756, n=817;
6. r=18, m=7;
7. m=5, n=9, r=11, k=9.

Вариант 21

1. m=998, n=1285;
2. m=1974, n=528;
3. n=474;
4. a=8, b=624, n=689;
5. r=33, m=5;
6. m=6, n=9, r=23, k=16.

Вариант 22

1. m=1934, n=725;
2. m=552, n=874;
3. n=375;
4. a=8, b=624, n=689;
5. r=25, m=6;
6. m=5, n=9, r=11, k=9.

Вариант 23

1. m=1778, n=994;
2. m=1236, n=935;
3. n=552;
4. a=5, b=520, n=583;
5. r=18, m=5;
6. m=6, n=9, r=23, k=16.

Вариант 24

1. m=844, n=1483;
2. m=472, n=844;
3. n=423;
4. a=24, b=936, n=1007;
5. r=44, m=7;
6. m=8, n=9, r=19, k=18.

Вариант 25

1. m=1379, n=254;
2. m=1116, n=975;
3. n=752;
4. a=5, b=520, n=583;
5. r=17, m=14;
6. m=6, n=9, r=23, k=16.

Вариант 26

1. m=844, n=1483;
2. m=472, n=844;
3. n=493;
4. a=24, b=936, n=907;
5. r=31, m=9;
6. m=8, n=9, r=19, k=18.

Вариант 27

1. m=2778, n=1194;
2. m=2236, n=1135;
3. n=374;
4. a=5, b=510, n=511;
5. r=19, m=11;
6. m=9, n=5, r=23, k=17.

Вариант 28

1. m=2844, n=1483;
2. m=3472, n=1844;
3. n=925;
4. a=24, b=936, n=807;
5. r=47, m=7;
6. m=8, n=7, r=17, k=18.

Вариант 29

1. m=2778, n=1994;
2. m=2236, n=335;
3. n=472;
4. a=5, b=520, n=583;
5. r=19, m=7;
6. m=6, n=9, r=23, k=17.

Вариант 30

1. m=844, n=2483;
2. m=2472, n=544;
3. n=583;
4. a=7, b=936, n=1007
5. r=43, m=11;
6. m=8, n=5, r=17, k=18.

Лабораторная работа №3
Криптографические системы с открытым ключом
(4 часа)

Цель работы: Изучить принципы работы криптосистемы с открытым ключом на основе алгоритма *RSA*.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Первые криптографические системы с открытым ключом появились в конце 1970-х годов. От классических алгоритмов они отличаются тем, что для шифрования данных используется один ключ (открытый), а для расшифрования – другой (секретный). Данные, зашифрованные открытым ключом, можно расшифровать только секретным ключом. Следовательно, открытый ключ может распространяться через обычные коммуникационные сети и другие открытые каналы. Таким образом, устраняется главный недостаток стандартных криптографических алгоритмов: необходимость использовать специальные каналы связи для распределения ключей. Разумеется, секретный ключ не может быть вычислен из открытого ключа.

В настоящее время лучшим криптографическим алгоритмом с открытым ключом считается *RSA* (по имени создателей: *Rivest, Shamir, Adelman*).

Наиболее важной частью алгоритма *RSA*, как и других алгоритмов с открытым ключом, является процесс создания пары открытый/секретный ключи. В *RSA* он состоит из следующих шагов.

1. Случайным образом выбираются два секретных простых числа, p и q, $p \neq q$.
2. Вычисляется $r = p * q$.
3. Вычисляется функция Эйлера $\varphi(r) = (p-1)*(q-1)$.
4. Выбираются открытый (Ko) и секретный (Kc) ключи, которые являются взаимно простыми с $\varphi(r)$ и удовлетворяют условию $(Ko * Kc) \bmod \varphi(r) = 1$. То есть, Ko является взаимообратным по модулю $\varphi(r)$ для Kc. Таким образом, ключом шифрования является пара значений (Ko, r). Ключом расшифрования является пара значений (Kc, r). Значение параметра r, так же как и значение ключа Ko, является общедоступной информацией в то время как значения параметров p, q и ключа Kc хранятся в секрете.

Чтобы зашифровать данные открытым ключом Ko, необходимо:

1) разбить исходный текст на блоки, каждый из которых может быть представлен в виде числа $M(i)$ в диапазоне $0 \ldots r-1$;

2) зашифровать последовательность чисел $M(i)$ по формуле:

$$C(i) = (M(i) Ko) \bmod r,$$

где последовательность чисел $C(i)$ представляет шифротекст.

Чтобы расшифровать эти данные секретным ключом K_C, необходимо выполнить следующие вычисления:

$$M(i) = (C(i) Kc) \bmod r.$$

В результате будет получено множество чисел $M(i)$, которые представляют собой исходный текст.

Приведем простой пример использования метода *RSA* для шифрования сообщения "*CAB*". Для простоты будем использовать малые числа (на практике используются намного большие числа).

1. Выберем p=3, q=11.
2. Вычислим r=3•11=33.
3. Вычислим $\varphi(r)=(p-1)\bullet(q-1)=20$.
4. Выберем секретный ключ K_C, который является взаимно простым с $\varphi(r)$, например K_C=3.
5. На основе K_C и $\varphi(r)$ вычислим открытый ключ *Ко*. Для этого можно использовать расширение алгоритма Евклида. Расширенный алгоритм Евклида позволяет вычислить x_1 и y_1, при которых выполняется равенство x_1•a+ y_1•b=d1, где d1=НОД(a,b). Если a и b – взаимнопростые и $a>b$, то y_1 является взаимообратным для b по модулю a. То есть, $y_1 \bullet b \bmod a = 1$.

Используя данный алгоритм можно вычислить *Ко* положив a равным $\varphi(r)$ и b равным K_C:

В соответствии с алгоритмом получаем $Ko = y_1 = 7$.

6. Представим шифруемое сообщение как последовательность целых чисел в диапазоне 2...28. Пусть букве '*A*' соответствует число 2, букве '*B*' – число 3, а букве '*C*' – число 4. Тогда сообщение "*CAB*" можно представить в виде последовательности чисел {5, 3, 4}. Зашифруем сообщение, используя открытый ключ *Ко*=7:

$C(1) = (57) \bmod 33 = 78125 \bmod 33 = 14$,

$C(2) = (37) \bmod 33 = 2187 \bmod 33 = 9$,

$C(3) = (47) \bmod 33 = 16384 \bmod 33 = 16$.

7. Для расшифровки полученного сообщения {14, 9, 16} с помощью секретного ключа *КС*=3, необходимо:

$M(1) = (143) \bmod 33 = 2744 \bmod 33 = 5$,

$M(2) = (93) \bmod 33 = 729 \bmod 33 = 3$,

$M(3) = (163) \bmod 33 = 4096 \bmod 33 = 4$.

Таким образом, в результате расшифрования сообщения получено исходное сообщение {5, 3, 4} («*CAB*»).

Криптостойкость алгоритма *RSA* основывается на предположении, что исключительно трудно определить секретный ключ по открытому, поскольку для этого необходимо решить задачу о существовании делителей целого числа, то есть, найти множители параметра r. Данная задача не имеет эффективного (полиномиального) решения. Вопрос существования эффективного алгоритма решения данной задачи является до настоящего времени открытым. Традиционные же методы для чисел, состоящих из 200 цифр (именно такие числа рекомендуется использовать), требуют выполнения огромного числа операций (порядка 1023).

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

2.1. Изучить теоретический материал по лабораторной работе.

2.2. Используя заданные в соответствии с вариантом значения *p*, *q* и закрытого ключа *Kc* вычислить открытый ключ *Ko* при помощи расширенного алгоритма Евклида и выполнить шифрование по алгоритму *RSA* открытым ключом (*Ko*) своей фамилии. Для представления букв в числовой форме использовать следующее соответствие: ‘А’ – 2, ‘Б’ – 3, ‘В’ – 4, ..., ‘Ё’ – 8, ..., ‘Я’ – 34.

2.3. Выполнить проверку правильности расшифрования полученных зашифрованных данных при помощи закрытого ключа *Kc*.

Варианты заданий

1. *p*=5, *q*=7, *Kc*=11.
2. *p*=17, *q*=5, *Kc*=7.
3. *p*=11, *q*=5, *Kc*=13.
4. *p*=7, *q*=11, *Kc*=19.
5. *p*=7, *q*=17, *Kc*=5.
6. *p*=3, *q*=17, *Kc*=23.
7. *p*=13, *q*=3, *Kc*=11.
8. *p*=5, *q*=13, *Kc*=19.
9. *p*=7, *q*=13, *Kc*=17.
10. *p*=3, *q*=19, *Kc*=7.
11. *p*=5, *q*=7, *Kc*=23.
12. *p*=17, *q*=5, *Kc*=19.
13. *p*=11, *q*=5, *Kc*=17.
14. *p*=7, *q*=11, *Kc*=13.
15. *p*=7, *q*=17, *Kc*=11.
16. *p*=3, *q*=17, *Kc*=13.
17. *p*=13, *q*=3, *Kc*=17.
18. *p*=5, *q*=13, *Kc*=11.
19. *p*=7, *q*=13, *Kc*=19.
20. *p*=3, *q*=19, *Kc*=13.
21. *p*=5, *q*=7, *Kc*=19.
22. *p*=17, *q*=5, *Kc*=23.
23. *p*=11, *q*=5, *Kc*=19.
24. *p*=7, *q*=11, *Kc*=17.
25. *p*=7, *q*=17, *Kc*=23.
26. *p*=3, *q*=17, *Kc*=11.
27. *p*=13, *q*=3, *Kc*=19.
28. *p*=5, *q*=13, *Kc*=17.
29. *p*=7, *q*=13, *Kc*=23.
30. *p*=3, *q*=19, *Kc*=11.

Лабораторная работа №4
Электронная цифровая подпись
(4 часа)

Цель работы: Изучить принципы формирования электронной цифровой подписи

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Функция хеширования

Функцией хеширования h называется преобразование данных, переводящее строку M произвольной длины в значение $m=h(M)$ (хеш-образ) некоторой фиксированной длины.

Хорошая хеш-функция должна удовлетворять следующим условиям:

1. Хеш-функция $h(M)$ должна быть чувствительна к любым изменениям входной последовательности M.

2. Для данного значения $h(M)$ должно быть невозможным нахожение значения M.

3. Для данного значения $h(M)$ должно быть невозможным нахождение $M' \neq M$ такого, что $h(M') = h(M)$.

4. Вероятность возникновения ситуации, называемой коллизией, когда для различных входных последовательностей M и M' совпадают значения их хеш-образов: $h(M) = h(M')$, должна быть чрезвычайно мала.

При построении хеш-образа входная последовательность M разбивается на блоки M_i фиксированной длины и обрабатывается поблочно по формуле:

$$H_i = f(H_{i-1}, M_i).$$

Хеш-значение, вычисленное в результате обработки последнего блока сообщения, становится хеш-значением (хеш-образом) всего сообщения.

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

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

где $n = p \cdot q$, p и q – большие простые числа, H_0 – произвольное начальное значение, M_i – i-й блок сообщения $M = \{M_1, M_2, \ldots, M_k\}$.

Пример. Вычислим хеш-образ для строки «ГГТУ». Для перехода от символов к числовым значениям будем использовать следующее соответствие: ‘А’ – 1; ‘Б’ – 2; ‘Г’ – 4; ‘Т’ – 20; ‘У’ – 21; ‘Я’ – 33.

Тогда сообщение M примет вид M ={4, 4, 20, 21}.

Выберем 2 простых числа p=17, q=19. Тогда модуль n=323.

Положим H_0=100.

$H_1=(H_0+M_1)^2 \bmod n=(100+4)^2 \bmod 323=10816 \bmod 323=157.$

$H_2=(H_1+M_2)^2 \bmod n=(157+4)^2 \bmod 323=81.$
$H_3=(81+20)^2 \bmod 323=188.$
$H_4=(188+21)^2 \bmod 323=\underline{\mathbf{76}}.$

Таким образом, $h(M)=H_4=76$.

1.2 Электронная цифровая подпись

Электронная цифровая подпись для электронных документов играет ту же роль, что и подпись, поставленная от руки в документах на бумаге: это данные, присоединяемые к передаваемому сообщению, подтверждающие, что владелец подписи составил или заверил это сообщение. Получатель сообщения с помощью цифровой подписи может проверить, что автором сообщения является именно владелец подписи и что в процессе передачи не была нарушена целостность полученных данных.

При разработке механизма цифровой подписи возникают следующие задачи:

- формирование подписи таким образом, чтобы её невозможно было подделать;
- обеспечение возможности проверки того, что подпись действительно принадлежит указанному субъекту;
- предотвращение отказа субъекта от своей подписи.

1.3 Классическая схема создания цифровой подписи

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

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

Получив подписанное сообщение (M', S), получатель должен выполнить следующие действия (принятое сообщение обозначено как M' по причине того, что оно могло быть преднамеренно либо случайно искажено в процессе передачи по каналу связи и может не совпадать с отправленным).

Вычислить хеш-образ m' сообщения M' при помощи хеш-функции h.

С использованием открытого ключа проверки подписи (K_o) извлечь хеш-образ m сообщения из цифровой подписи S.

Сравнить вычисленное значение *m'* с извлеченным из цифровой подписи значением хеш-образа *m*. Если хеш-образы совпадают, то подпись признается подлинной.

1.4 Алгоритм цифровой подписи RSA

Первой и наиболее известной во всем мире конкретной системой электронной цифровой подписи стала система *RSA*, математическая схема которой была разработана в 1977 г. в Массачусетском технологическом институте США.

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель сообщения (документа) выбирает два больших простых числа p и q, а затем находит их произведение

$$r = p \cdot q$$

и значение функции Эйлера от данного произведения

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

Далее отправитель вычисляет значение K_o из условий:

$$K_o < \varphi(r),\ НОД(K_o, \varphi(r)) = 1$$

и значение K_c из условий:

$$K_c < \varphi(r),\ K_o \cdot K_c = 1 \bmod \varphi(r).$$

Пара значений (K_o, r) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Значение K_c сохраняется автором как секретный ключ подписи.

Обобщенная схема формирования и проверки цифровой подписи RSA показана на рисунке 1.

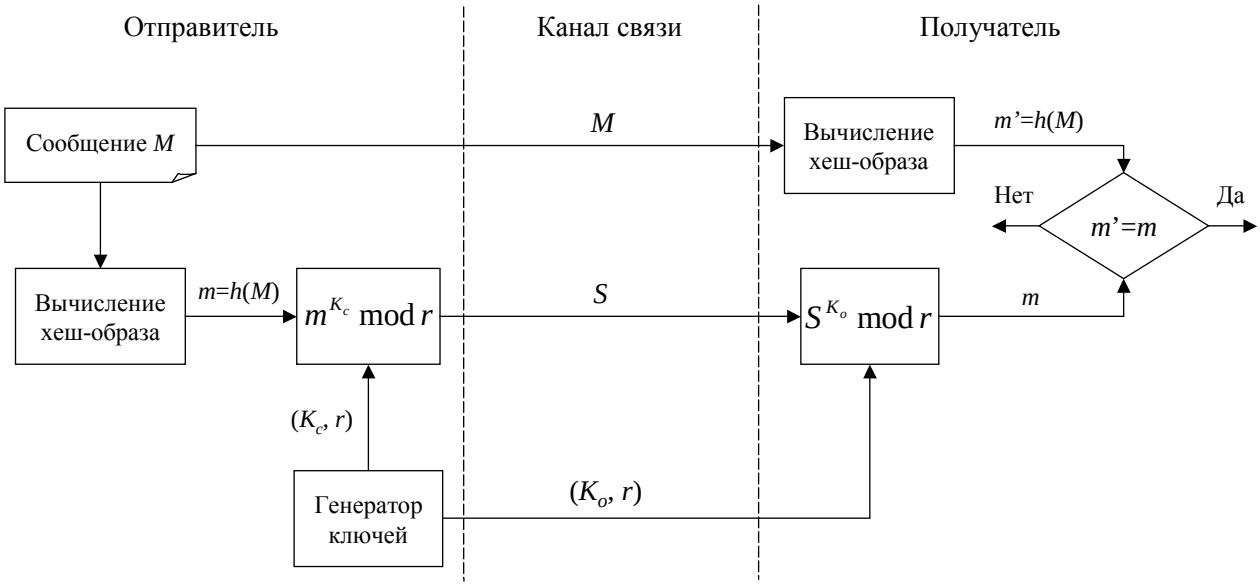


Рисунок 1 – Обобщенная схема цифровой подписи *RSA*

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M сжимают с помощью хеш-функции h в целое число m:

$m = h(M)$.

Затем вычисляют цифровую подпись S под электронным документом M, на основе хеш-образа m и секретного значения K_c:

$$S = m^{K_c} \bmod r .$$

Для возведения в степень можно воспользоваться алгоритмом быстрого возведения в степень по модулю, позволяющим вычислить

$x = a^z \bmod n$.

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

После приема пары (M', S) получатель вычисляет хеш-образ сообщения M' двумя различными способами. Прежде всего, он восстанавливает хеш-образ m, применяя криптографическое преобразование подписи S с использованием открытого ключа K_o:

$$m = S^{K_o} \bmod r .$$

Кроме того, он находит результат хеширования m' принятого сообщения M' с помощью такой же хеш-функции h:

$$m' = h(M).$$

Если вычисленные значения совпадают, то есть:

$$S^{K_o} \bmod r = h(M'),$$

то получатель признает пару (M', S) подлинной. Фальсификиция сообщения при его передаче по каналу связи возможна только при получении злоумышленником секретного ключа K_c либо за счет проведения успешной атаки против хеш-функции. При использовании достаточно больших значений p и q определение секретного значения K_c по открытому ключу (K_o,r) является чрезвычайно трудной задачей, соответствующей по сложности разложению модуля r на множители. Используемые в реальных приложениях хеш-функции обладают характеристиками, делающими атаку против цифровой подписи практически не осуществимой. Пример – хеш-функция SHA-1, принятая в США в качестве стандарта в 1995 году, формирующая 160-битовый хеш-образ при обработке сообщения блоками по 512 бит. Вероятность коллизии при использовании данной хеш-функции составляет 2^{-160} или приблизительно $6.84 \cdot 10^{-49}$.

2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Найти хеш-образ своей фамилии, используя хеш-функцию
$$H_i = (H_{i-1} + M_i)^2 \bmod n, \text{ где } n = p \cdot q.$$
2. Показать как меняется хеш-образ при изменении одной из букв в фамилии (вычислить $h(M')$, где M' – фамилия с одной измененной буквой).
3. Показать (вычислить $h(M'')$) как меняется хеш-образ при перестановке любых двух букв в фамилии.
4. Используя полученный ранее хеш-образ вычислить электронную цифровую подпись для своей фамилии по схеме *RSA*. При вычислении подписи использовать алгоритм быстрого возведения в степень по модулю.

Варианты заданий

1. p=13, q=17	11. p=17, q=19	21. p=23, q=19
2. p=23, q=13	12. p=7, q=23	22. p=17, q=13
3. p=19, q=11	13. p=7, q=29	23. p=19, q=17
4. p=17, q=23	14. p=7, q=19	24. p=13, q=23
5. p=19, q=13	15. p=7, q=31	25. p=23, q=11
6. p=11, q=29	16. p=7, q=37	26. p=29, q=13
7. p=19, q=23	17. p=5, q=31	27. p=23, q=7
8. p=11, q=23	18. p=5, q=37	28. p=31, q=7
9. p=11, q=17	19. p=5, q=29	29. p=29, q=17
10. p=13, q=29	20. p=29, q=11	30. p=23, q=29