МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

О.А. ГУЛЯЄВА

Методичні вказівки до виконання практичних робіт з дисципліни «Захист інформації» для студентів напряму 6.020100-документознавство та інформаційна діяльність

Дніпропетровськ
2015

Содержание

Тема 1. Основы криптографирования

Общие положения

В качестве информации, подлежащей *шифрованию* и *дешифрованию,* будут рассматриваться *тексты*, построенные на некотором *алфавите*. Под этими терминами понимается следующее.

Алфавит - конечное множество используемых для кодирования информации знаков.

Текст - упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных информационных системах (ИС) можно привести следующие:

• алфавит Z_{33} - 32 буквы русского алфавита и пробел;
• алфавит Z_{44} – 43 буквы русского алфавита, знаки препинания и пробела;
• алфавит Z_{256} – символы, входящие в стандартные коды ASCII и КОИ -8;
• бинарный алфавит – $Z_2 = \{0,1\}$;
• восьмеричный алфавит – $Z_8 = \{0,1,2,3,4,5,6,7\}$;
• шестнадцатеричный алфавит – $Z_{16} = \{0,1,2,3,4,5,6,7,8,9,10,11, 12,13, 14, 15\}$;

Простые операции над двоичным кодом, обращаемые функции

Напомним, что символы представляют собой целые числа без знака лежачие в диапазоне значений 0 – 255, т.е. представляют собой тип данных – байт.

В двоичной системе символы и числа представляют собой последовательность 0 и 1, содержащую 8 разрядов. Например, в системе ASCII (алфавит Z256) двоичный код числа 5 будет равен 00110101, его шестнадцатеричное представление 35, а порядковый номер в таблице равен 53. Аналогично символ Ж русского алфавита имеет двоичный код 1100 0110, шестнадцатеричное представление С6, а порядковый номер таблицы равен 198.

Примечание. *Порядковый номер символа в таблице это десятичный код символа.*

В таблицах 1 и 2 приведены двоичные коды символов и их щестнадцатиричное представление в системе ASCII.

Над числами (кодами символов) могут выполняться различные действия. Простые числа можно складывать, вычитать, умножать, делить и возводить в степень.

Особый интерес для криптографирования представляют логические функции, выполняющие поразрядные действия с двоичными кодами.

К таким функциям относятся:

AND – логическое И (умножение);
OR - логическое ИЛИ (сложение);

XOR – функция ИСКЛЮЧИТЕЛЬНО ИЛИ (сумма по модулю 2);
NOT – отрицание НЕ (инверсия).

Следует заметить, что последние две функции являются обратимыми.

Рассмотрим выполнение приведенных функций.

1. Выполним операции AND и OR над кодами символов A и B латинского алфавита:

	0100 0001 – A		0100 0001 – A
AND		OR	
	0100 0010 – B		0100 0010 - B
	----------		----------
	0100 0000 - @		0100 0011 – C

2. Инверсия кода символа A вернет код 1011 1110, который будет соответствовать символу **ï**. Если выполнить повторно операцию инверсии NOT(**ï**), то получим код 0100 0001, соответствующий символу A.

Таблица 1 (код 0 - 127)

D	H	B	S	D	H	B	S	D	H	B	S	D	H	B	S
0	H 00	B 0000 0000	(CS)	32	H 20	B 0010 0000	spc	64	H 40	B 0100 0000	@	96	H 60	B 0110 0000	`
1	H 01	B 0000 0001	(CS)	33	H 21	B 0010 0001	!	65	H 41	B 0100 0001	A	97	H 61	B 0110 0001	a
2	H 02	B 0000 0010	(CS)	34	H 22	B 0010 0010	"	66	H 42	B 0100 0010	B	98	H 62	B 0110 0010	b
3	H 03	B 0000 0011	(CS)	35	H 23	B 0010 0011	#	67	H 43	B 0100 0011	C	99	H 63	B 0110 0011	c
4	H 04	B 0000 0100	(CS)	36	H 24	B 0010 0100	$	68	H 44	B 0100 0100	D	100	H 64	B 0110 0100	d
5	H 05	B 0000 0101	(CS)	37	H 25	B 0010 0101	%	69	H 45	B 0100 0101	E	101	H 65	B 0110 0101	e
6	H 06	B 0000 0110	(CS)	38	H 26	B 0010 0110	&	70	H 46	B 0100 0110	F	102	H 66	B 0110 0110	f
7	H 07	B 0000 0111	(CS)	39	H 27	B 0010 0111	(')	71	H 47	B 0100 0111	G	103	H 67	B 0110 0111	g
8	H 08	B 0000 1000	(CS)	40	H 28	B 0010 1000	(	72	H 48	B 0100 1000	H	104	H 68	B 0110 1000	h
9	H 09	B 0000 1001	(CS)	41	H 29	B 0010 1001	)	73	H 49	B 0100 1001	I	105	H 69	B 0110 1001	i
10	H 0A	B 0000 1010	(CS)	42	H 2A	B 0010 1010	*	74	H 4A	B 0100 1010	J	106	H 6A	B 0110 1010	j
11	H 0B	B 0000 1011	(CS)	43	H 2B	B 0010 1011	+	75	H 4B	B 0100 1011	K	107	H 6B	B 0110 1011	k
12	H 0C	B 0000 1100	(CS)	44	H 2C	B 0010 1100	,	76	H 4C	B 0100 1100	L	108	H 6C	B 0110 1100	l
13	H 0D	B 0000 1101	(CS)	45	H 2D	B 0010 1101	-	77	H 4D	B 0100 1101	M	109	H 6D	B 0110 1101	m
14	H 0E	B 0000 1110	(CS)	46	H 2E	B 0010 1110	.	78	H 4E	B 0100 1110	N	110	H 6E	B 0110 1110	n
15	H 0F	B 0000 1111	(CS)	47	H 2F	B 0010 1111	/	79	H 4F	B 0100 1111	O	111	H 6F	B 0110 1111	o
16	H 10	B 0001 0000	(CS)	48	H 30	B 0011 0000	0	80	H 50	B 0101 0000	P	112	H 70	B 0111 0000	p
17	H 11	B 0001 0001	(CS)	49	H 31	B 0011 0001	1	81	H 51	B 0101 0001	Q	113	H 71	B 0111 0001	q
18	H 12	B 0001 0010	(CS)	50	H 32	B 0011 0010	2	82	H 52	B 0101 0010	R	114	H 72	B 0111 0010	r
19	H 13	B 0001 0011	(CS)	51	H 33	B 0011 0011	3	83	H 53	B 0101 0011	S	115	H 73	B 0111 0011	s
20	H 14	B 0001 0100	(CS)	52	H 34	B 0011 0100	4	84	H 54	B 0101 0100	T	116	H 74	B 0111 0100	t
21	H 15	B 0001 0101	(CS)	53	H 35	B 0011 0101	5	85	H 55	B 0101 0101	U	117	H 75	B 0111 0101	u
22	H 16	B 0001 0110	(CS)	54	H 36	B 0011 0110	6	86	H 56	B 0101 0110	V	118	H 76	B 0111 0110	v
23	H 17	B 0001 0111	(CS)	55	H 37	B 0011 0111	7	87	H 57	B 0101 0111	V	119	H 77	B 0111 0111	w
24	H 18	B 0001 1000	(CS)	56	H 38	B 0011 1000	8	88	H 58	B 0101 1000	X	120	H 78	B 0111 1000	x
25	H 19	B 0001 1001	(CS)	57	H 39	B 0011 1001	9	89	H 59	B 0101 1001	Y	121	H 79	B 0111 1001	y
26	H 1A	B 0001 1010	(CS)	58	H 3A	B 0011 1010	:	90	H 5A	B 0101 1010	Z	122	H 7A	B 0111 1010	z
27	H 1B	B 0001 1011	(CS)	59	H 3B	B 0011 1011	;	91	H 5B	B 0101 1011	[	123	H 7B	B 0111 1011	{
28	H 1C	B 0001 1100	(CS)	60	H 3C	B 0011 1100	<	92	H 5C	B 0101 1100	\	124	H 7C	B 0111 1100	|
29	H 1D	B 0001 1101	(CS)	61	H 3D	B 0011 1101	=	93	H 5D	B 0101 1101	]	125	H 7D	B 0111 1101	}
30	H 1E	B 0001 1110	(CS)	62	H 3E	B 0011 1110	>	94	H 5E	B 0101 1110	^	126	H 7E	B 0111 1110	~
31	H 1F	B 0001 1111	(CS)	63	H 3F	B 0011 1111	?	95	H 5F	B 0101 1111	_	127	H 7F	B 0111 1111	(CS)

Таблица 2 (код 128 - 255)

D	H	B	S	D	H	B	S	D	H	B	S	D	H	B	S
128	H 80	B 1000 0000	(CS)	160	H A0	B 1010 0000	spc	192	H C0	B 1100 0000	**А**	224	H E0	B 1110 0000	**а**
129	H 81	B 1000 0001	(CS)	161	H A1	B 1010 0001	**Ў**	193	H C1	B 1100 0001	**Б**	225	H E1	B 1110 0001	**б**
130	H 82	B 1000 0010	(CS)	162	H A2	B 1010 0010	**ў**	194	H C2	B 1100 0010	**В**	226	H E2	B 1110 0010	**в**
131	H 83	B 1000 0011	(CS)	163	H A3	B 1010 0011	**J**	195	H C3	B 1100 0011	**Г**	227	H E3	B 1110 0011	**г**
132	H 84	B 1000 0100	(CS)	164	H A4	B 1010 0100	¤	196	H C4	B 1100 0100	**Д**	228	H E4	B 1110 0100	**д**
133	H 85	B 1000 0101	(CS)	165	H A5	B 1010 0101	**Ґ**	197	H C5	B 1100 0101	**Е**	229	H E5	B 1110 0101	**е**
134	H 86	B 1000 0110	(CS)	166	H A6	B 1010 0110	¦	198	H C6	B 1100 0110	**Ж**	230	H E6	B 1110 0110	**ж**
135	H 87	B 1000 0111	(CS)	167	H A7	B 1010 0111	**§**	199	H C7	B 1100 0111	**З**	231	H E7	B 1110 0111	**з**
136	H 88	B 1000 1000	(CS)	168	H A8	B 1010 1000	**Ё**	200	H C8	B 1100 1000	**И**	232	H E8	B 1110 1000	**и**
137	H 89	B 1000 1001	(CS)	169	H A9	B 1010 1001	©	201	H C9	B 1100 1001	**Й**	233	H E9	B 1110 1001	**й**
138	H 8A	B 1000 1010	(CS)	170	H AA	B 1010 1010	**Є**	202	H CA	B 1100 1010	**К**	234	H EA	B 1110 1010	**к**
139	H 8B	B 1000 1011	(CS)	171	H AB	B 1010 1011	«	203	H CB	B 1100 1011	**Л**	235	H EB	B 1110 1011	**л**
140	H 8C	B 1000 1100	(CS)	172	H AC	B 1010 1100	¬	204	H CC	B 1100 1100	**М**	236	H EC	B 1110 1100	**м**
141	H 8D	B 1000 1101	(CS)	173	H AD	B 1010 1101	-	205	H CD	B 1100 1101	**Н**	237	H ED	B 1110 1101	**н**
142	H 8E	B 1000 1110	(CS)	174	H AE	B 1010 1110	®	206	H CE	B 1100 1110	**О**	238	H EE	B 1110 1110	**о**
143	H 8F	B 1000 1111	(CS)	175	H AF	B 1010 1111	**Ї**	207	H CF	B 1100 1111	**П**	239	H EF	B 1110 1111	**п**
144	H 90	B 1001 0000	(CS)	176	H B0	B 1011 0000	°	208	H D0	B 1101 0000	**Р**	240	H F0	B 1111 0000	**р**
145	H 91	B 1001 0001	(CS)	177	H B1	B 1011 0001	±	209	H D1	B 1101 0001	**С**	241	H F1	B 1111 0001	**с**
146	H 92	B 1001 0010	(CS)	178	H B2	B 1011 0010	**I**	210	H D2	B 1101 0010	**Т**	242	H F2	B 1111 0010	**т**
147	H 93	B 1001 0011	(CS)	179	H B3	B 1011 0011	**i**	211	H D3	B 1101 0011	**У**	243	H F3	B 1111 0011	**у**
148	H 94	B 1001 0100	(CS)	180	H B4	B 1011 0100	**ґ**	212	H D4	B 1101 0100	**Ф**	244	H F4	B 1111 0100	**ф**
149	H 95	B 1001 0101	(CS)	181	H B5	B 1011 0101	**µ**	213	H D5	B 1101 0101	**Х**	245	H F5	B 1111 0101	**х**
150	H 96	B 1001 0110	(CS)	182	H B6	B 1011 0110	¶	214	H D6	B 1101 0110	**Ц**	246	H F6	B 1111 0110	**ц**
151	H 97	B 1001 0111	(CS)	183	H B7	B 1011 0111	·	215	H D7	B 1101 0111	**Ч**	247	H F7	B 1111 0111	**ч**
152	H 98	B 1001 1000	(CS)	184	H B8	B 1011 1000	**ё**	216	H D8	B 1101 1000	**Ш**	248	H F8	B 1111 1000	**ш**
153	H 99	B 1001 1001	(CS)	185	H B9	B 1011 1001	**№**	217	H D9	B 1101 1001	**Щ**	249	H F9	B 1111 1001	**щ**
154	H 9A	B 1001 1010	(CS)	186	H BA	B 1011 1010	**є**	218	H DA	B 1101 1010	**Ъ**	250	H FA	B 1111 1010	**ъ**
155	H 9B	B 1001 1011	(CS)	187	H BB	B 1011 1011	»	219	H DB	B 1101 1011	**Ы**	251	H FB	B 1111 1011	**ы**
156	H 9C	B 1001 1100	(CS)	188	H BC	B 1011 1100	**j**	220	H DC	B 1101 1100	**Ь**	252	H FC	B 1111 1100	**ь**
157	H 9D	B 1001 1101	(CS)	189	H BD	B 1011 1101	**S**	221	H DD	B 1101 1101	**Э**	253	H FD	B 11111 1101	**э**
158	H 9E	B 1001 1110	(CS)	190	H BE	B 1011 1110	**s**	222	H DE	B 1101 1110	**Ю**	254	H FE	B 1111 1110	**ю**
159	H 9F	B 1001 1111	(CS)	191	H BF	B 1011 1111	**ї**	223	H DF	B 1101 1111	**Я**	255	H FF	B 1111 1111	**я**

Функция XOR является комплексной функцией, содержащей простые функции AND, OR и NOT. Эта функция является обратимой и представляет наибольший интерес в системах шифрования. Наиболее часто она применяется в гаммах ключей и шифров.

Рассмотрим на примерах прямое и обратное действие указанной функции.

Пример 1. Выполним функцию XOR между символами A и B латинского алфавита. Затем выполним эту же функцию между полученным результатом и символом B. Оценим результат.

0100 0001 – A
XOR
0100 0010 – B

0000 0011 – (CS)
XOR
0100 0010 - B

0000 0011 – (CS)

0100 0001 – A

В результате выполнения первого действия получим код управляющего символа, который не отображается. В результате второго действия будет возвращен код символа А из кода управляющего символа.

Данный пример наглядно демонстрирует простейшую операцию шифрования и дешифрования, выполняемую в эвм.

Пример 2. Рассмотрим пример шифрования слова “СЕКРЕТ” некоторой цифровой последовательностью – 567890, являющейся ключом шифра.

С	Е	К	Р	Е	Т
1101 0001	1100 0101	1100 1010	1101 0000	1100 0101	1101 0010
5	6	7	8	9	0
0011 0101	0011 0110	0011 0111	0011 1000	0011 1001	0011 0000
1110 0100	1111 0011	1111 1101	1110 1000	1111 1110	1110 0010
д	у	э	и	ю	в
228	243	253	232	254	226

В результате шифрования получена не смысловой набор символов прописных букв русского алфавита “дуэиюв”. Далее усложним шифр заменой символов на их эквивалентные десятичные коды (номера символов). Данная цифровая последовательность может быть сохранена, либо передана получателю ввиде текстового или бинарного файла, например, фрагмента изображения.

Декодирование файла выполняется в обратной последовательности.

Пример 3. Рассмотрим другой вариант криптографирования слова “СЕКРЕТ”. В качестве ключа выберем две цифры, например 5 и 6. Последовательно выполним функцию XOR каждого символа шифруемого слова сначала с цифрой 5, а затем с цифрой 6.

С	Е	К	Р	Е	Т
1101 0001	1100 0101	1100 1010	1101 0000	1100 0101	1101 0010
5	5	5	5	5	5
0011 0101	0011 0101	0011 0101	0011 0101	0011 0101	0011 0101
1110 0100	1111 0000	1111 1111	1110 0101	1111 0000	1110 0111
6	6	6	6	6	6
0011 0110	0011 0110	0011 0110	0011 0110	0011 0110	0011 0110
1101 0010	1100 0110	1100 1001	1101 0011	1100 0110	1101 0001
Т	Ц	Й	У	Ж	С

В результате шифрования получен набор символов “ТЦЙУЖС”.

Программно данный алгоритм реализуется весьма просто. Операция шифрования выполняется в цикле путем последовательной подстановки каждого символа ключа в каждый шаг цикла. Число циклов шифрования равно количеству символов в ключе.

Приведенный метод шифрования является составной частью ряда алгоритмов криптографирования, например установка пароля в архиваторе ZIP32.

Как видно из алгоритма, чем длинее пароль, тем труднее вскрыть указанный шифр. Однако у данного метода имеется большой недостаток – это время шифрования и дешифрования больших файлов пропорционально увеличиваемое в соответствии с длиной пароля.

Тема 2. Маршрутная транспозиция

К классу перестановка относится шифр ***маршрутная транспозиция*** и его вариант ***постолбцовая транспозиция***. В каждом из них в данный прямоугольник размерами ***[n *m]*** вписывается сообщение заранее обусловленным способом, а столбцы нумеруются или обычным порядком следования, или в порядке следования букв ключа – буквенного ключевого слова.

Пример 1. Зашифруем фразу “Дела давно минувших дней, преданья старины глубокой”, используя для этого два прямоугольника 6*8.

В первом прямоугольнике столбцы нумеруются в обычном порядке следования – слева направо, а во втором – в порядке следования букв слова “Пушкин”. Используя расположение букв этого ключа в алфавите, получим набор чисел [4 5 6 2 1 3]:

4 5 6 2 1 4

Ключ: ПУШКИН

1	2	3	4	5	6
д	е	л	а	д	а
в	н	о	м	и	н
у	в	ш	и	х	д
н	е	й	п	р	е
д	а	н	ь	я	с
т	а	р	и	н	ы
г	л	у	б	о	к
о	й	а	б	в	г

4	5	6	2	1	3
д	е	л	а	д	а
в	н	о	м	и	н
у	в	ш	и	х	д
н	е	й	п	р	е
д	а	н	ь	я	с
т	а	р	и	н	ы
г	л	у	б	о	к
о	й	а	б	в	г

В первом случае получим шифрованный текст, если будем выписывать буквы очередного столбца в порядке следования столбцов (прямом или обратном), во втором, – если будем выписывать буквы столбца в порядке

следования букв ключа. Пустые клетки прямоугольника последовательно заполняются символами алфавита. Таким образом, будем иметь:

1) двундтго енвеаалй лошйнруа амипьибб дихрянов андесыкг; *(порядок)*
2) дихрянов амипьибб андесыкг двундтго енвеаалй лошйнруа. *(ключ)*

Видно, что образованные группы символов и количество групп подсказывают размеры прямоугольников. С целью усложнения дешифрования достаточно слить все символы шифрованного сообщения.

В первом случае для расшифрования сообщения необходимо знать только размеры прямоугольника и порядок нумерации столбцов, если последний применяется. Во втором случае необходимо помимо знания размеров прямоугольника иметь ключ дешифрования, который позволит определить порядок нумерации столбцов.

Тема 3. Таблица Виженера

В процессе шифрования и дешифрования иногда используется *таблица Виженера*, которая устроена следующим образом: в первой строке выписывается весь алфавит, в каждой следующей осуществляется циклический сдвиг на одну букву. Так получается квадратная таблица, число строк которой равно числу букв в алфавите. Чтобы зашифровать какое - либо сообщение, поступают следующим образом. Выбирается слово – лозунг и подписывается с повторением над буквами сообщения.

Чтобы получить шифрованный текст, находят очередной знак лозунга, начиная с первого в вертикальном алфавите, а ему соответствующий знак сообщения в горизонтальном.

Пример 2. Таблица 1, составлена (без букв Ё и Ъ).

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

Выбираем лозунг – “математика”. Находим столбец, отвечающий букве "м" лозунга, а затем строку, соответствующую букве "к". На пересечении выделенных столбца и строки находим букву "ц". Так продолжая дальше, получаем весь шифрованный текст:

м а т е м а т и к а м а т е м а т и к а м а т е м а *(лозунг)*

к р и п т о г р а ф и я с е р ь е з н а я н а у к а *(фраза)*

ц р ь ф я о х ш к ф ф я д к э ь ч п ч а л н т ш ц а *(криптограмма)*

К сообщению можно применять несколько систем шифрования.

Дешифрование выполняется в обратной последовательности.

Тема 4. Модифицированный шифр Цезаря

Аббат Тритемеус – автор первой печатной книги о тайнописи (1518 г.) – предложил несколько шифров и среди них шифр, который можно считать усовершенствованием шифра Цезаря.

Этот шифр устроен следующим образом. Все буквы алфавита нумеруются по порядку (от 1 до 31 в русском варианте). Затем выбирают какое - нибудь слово, называемое "ключом ", и подписывают под сообщением с повторением.

Чтобы получить шифрованный текст, складывают номер очередной буквы с номером соответствующей буквы ключа. Если полученная сумма больше 31, то из нее вычитают 31. В результате получают последовательность чисел от 1 до 31. Вновь заменяя числа этой последовательности соответствующими буквами, получают шифрованный текст.

Разбиваем этот текст на группы одной длины, получают шифрованное сообщение.

Пример 1. Выбираем ключевое слово "Пособие". Составляем сообщение "сессия начинается в конце семестра"

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я									
21	22	23	24	25	26	27	28	29	30	31									

с е с с и я н а ч и н а е т с я в к о н ц е с е м е с т р а
п о с о б и е п о с о б и е п о с о б и е п о с о б и е п о

Шифруем, разбиваем текст на группы длины *6,* и получаем шифрованное сообщение :

в ф д б к и у р з ы э в о ш в о ф щ р ц э х б ч ы з ь ш б п

Для дешифрования сообшения используется следующий алгоритм:

Складываются символы шифротекста с числом 31, а затем из суммы вычитается значение символа ключа. Если полученное значение не превышает 31 то результат будет номером символа сообщения. Если полученное значение больше 31 то выполняется вычитание из символа криптограммы соответствующего символа ключа.

1. В П
   3 + 31 - 16 = 18 → С
2. Ф О Ф О
   21 + 31 – 15 = 37. Так, как 37 > 31 то выполняем 21 – 15 = 6 → Е
3. Д С
   5 + 31 – 18 = 18 → С
4. Б О
   2 + 31 – 15 = 18 → С
5. К Б К Б
   11 + 31 - 2 = 40. Так, как 40 > 31 то выполняем 11 – 2 = 9 → И

Дальнейшие действия выполняются в той – же последовательности.

Для полного алфавита, содержащего 33 буквы, чтобы получить шифрованный текст, складывают номер очередной буквы с номером соответствующей буквы ключа.

Если полученная сумма больше 33, то из нее вычитают 33. В результате получают последовательность чисел от 1 до 33. Вновь заменяя числа этой последовательности соответствующими буквами, получают шифрованный текст. Для удобства, разбивая этот текст на группы одной длины (например , по *5),* получают шифрованное сообщение.

Если под ключом шифра понимать однобуквенное слово “В” (в русском варианте), то мы получим шифр Цезаря.

Пример 2. Для сообщения из примера 1, получим :

с	е	с	с	и	я
18	6	18	18	9	31
Б	Б	Б	Б	Б	Б
2	2	2	2	2	2

20	8	20	20	11	2
У	З	У	У	К	Б

Тема 5. Одноразовый блокнот

Целью настоящего занятия является изучение алгоритмов шифрования и дешифрования с использованием выданных листов одноразового блокнота. Работа выполняется по – парно двумя абонентами (студентами) на одинаковых листах одноразового блокнота.

Почти все используемые на практике шифры характеризуются как условно надежные, поскольку они могут быть раскрыты в принципе при наличии неограниченных вычислительных возможностей. Абсолютно надежные шифры нельзя разрушить даже при наличии неограниченных вычислительных возможностей. Доказательство существования и единственности абсолютно надежного шифра получил К. Шеннон с помощью разработанного им теоретико - информационного метода исследования шифров.

Таким образом, единственный абсолютно надежный шифр, который используется на практике, это так называемый *одноразовый блокнот*, в основе которого лежит та же идея, что и шифре Цезаря. Рассмотрим его основную идею. Занумеровав все символы расширенного алфавита Z_{44} числами от 0 до 43, можно рассматривать любой передавамый текст, как последовательность $\{a_n\}$ чисел множества **A** ={**0,1,2,…,4**3}. Имея случайную последовательность $\{c_n\}$ из чисел множества **A** той же длины что и передаваемый текст (ключ), складываем по модулю 44 число a_n передаваемого текста с соответствующим числом c_n ключа получим последовательность $\{b_n\}$ знаков шифрованного текста.

$$a_n+c_n \equiv b_n (\mathbf{mod}\ 44),\ 0 \le b_n \le 43,$$

Чтобы получить передаваемый текст, можно воспользоваться тем же ключом:

$$a_n \equiv b_n - c_n (\mathbf{mod}\ 44),\ 0 \le a_n \le 43.$$

У двух абонентов, находящихся в секретной переписке, имеются два одинаковых блокнота, составленных из отрывных страниц, на каждой из которых напечатана таблица со случайными числами или буквами, т.е. случайная последовательность чисел из множества **A**.

Таблица имеет только две копии: одна используется отправителем, другая – получателем. Отправитель свой текст шифрует указанным выше способом при помощи первой страницы блокнота. Зашифровав сообщение и отправив его второму абоненту, он уничтожает использованную страницу. Получатель шифрованного текста расшифровывает его и также уничтожает использованный лист блокнота.

Нетрудно видеть, что одноразовый шифр не раскрываем в принципе, так как символ в тексте может быть заменен любым другим символом и этот выбор совершенно случаен.

Заметим, что одноразовый блокнот, он же “Русский Блокнот” эффективно использовался Советской разведкой в период Великой Отечественной Войны.

Также он остается по – прежнему актуальным в армиях и спецслужбах стран СНГ, т.к. легко и эффективно аппаратно реализуется. Передача информации осуществляется по свободным каналам связи и другим средствам коммуникаций в виде передаваемых групп цифр.

Чтобы изготовить страницу одноразового блокнота из упорядоченных множеств (рис.1) используются специальные генераторы случайных цифровых последовательностей. Вариант результата работы генератора для одной строки приведен на (рис. 2).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
0	1	2	3	4	5	6	7	8	9	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	.	,

Рис.1. Исходная таблица

5	16	22	7	32	1	24	8	31	21	6	33	19	23	3	37	30	15	42	9	41	0	43	18	17	40	39	10	34	4	35	20	25	14	2	27	36	26	12	38	28	13	29	11
0	1	2	3	4	5	6	7	8	9	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	.	,

Рис. 2. Созданная таблица

Как видно из таблицы (рис. 2) генератор воздействовал только на цифровую последовательность.

В реальных условиях генератор воздействует и на символьную последовательность, что приводит к случайному распределению символов и во второй строке таблицы.

Тема 6. Фундаментальные алгоритмы криптографирования

Два самых фундаментальных алгоритма – это алгоритм деления и алгоритм Эвклида. Алгоритм деления предназначен для вычисления неполного частного и остатка при делении двух целых чисел. Алгоритм Эвклида вычисляет наибольший общий делитель двух целых чисел.

Алгоритмы деления

Задача сводится к нахождению частного и остатка от деления двух положительных целых чисел, например:

$$\begin{array}{r|l} 1234 & 54 \\ \hline 108 & 22 \\ \hline 154 & \\ 108 & \\ \hline 46 & \end{array}$$

В данном примере мы делим 1234 на 54. Неполное частное равно 22, а остаток равен 46. Соответственно делимое равно 1234, а делитель равен 54.

В общем случае ввод алгоритма деления состоит из двух положительных чисел ***a*** и ***b***. Деля ***a*** на ***b*** мы получаем при выводе числа ***q*** и ***r***, которые связаны с ***a*** и ***b*** следующим образом:

$$a = bq + r \text{ и } 0 \leqslant r < b. \qquad 1)$$

Где: ***q*** – это неполное целое, а ***r*** – остаток от деления.

Реализация алгоритма деления имеет вид:

Ввод: значения натуральных чисел a и b.
Вывод: положительные числа ***q*** и ***r***, для которых выполнено равенство 1).

Шаг 1. Присвоить ***Q = 0*** и ***R = a***.
Шаг 2. Если ***R < b***, то частное ***q = Q***, а остаток ***r = R***. Перейти к выходу. В противном случае перейти к шагу 3.
Шаг 3. Если ***R*** ≥ ***b***, то вычесть ***b*** из ***R***, увеличить ***Q*** на 1 возвратится к шагу 2.

Примечание: *В процессе функционирования алгоритма шаги 2 и 3 могут повторяться многократно – т.е. они образуют цикл. Значения переменных Q и R будут изменяться от цикла к циклу.*

Если ***a > b,*** то после первого прохода мы получим ***Q = 1*** и ***R = a – b***.

Если ***a – b*** ≥ ***b***, то выполняется шаг 3 еще раз. В следующем шаге ***Q = 2*** и ***R = a – b***, и т.д.

Заметим, что результате последовательного применения шага 3 получаем последовательность значений переменной ***R***:

Начальное значение	1-й цикл	2-й цикл	3-й цикл	...
a	$a-b$	$a-2b$	$a-3b$	...

Это убывающая последовательность целых чисел. Поскольку количество чисел между ***a*** и 0 конечно, последовательность попадет в число, меньшее ***b***. В этом случае на шаге 2 работа остановится, и алгоритм выводит значения переменных ***R*** и ***Q***.

Вывод*: алгоритм деления приводит к теореме, состоящих из двух утверждений: неполное частное и остаток от деления двух натуральных чисел всегда существует и они единственны.*

Теорема деления

Пусть a и b - натуральные числа. Тогда существует единственная пара положительных целых чисел q и r таких, что a = bq + r и 0 ≤ r < b.

Теорема содержит два утверждения про числа q и r. Во – первых они существуют, во – вторых они единственны.

Алгоритм Эвкклида

Определим, что целое число ***b*** делит целое число ***a***, если существует еще одно целое число ***c*** такое, что ***a = bc***. В этом случае говорят, что ***b*** является делителем или множителем числа ***a***, а ***a*** в свою очередь, - кратным числа ***b***.

Пусть ***a*** и ***b*** – натуральные числа. *Наибольший общий делитель* (НОД) чисел ***a*** и ***b*** – это наибольшее целое число ***d***, на которое и ***a*** и ***b*** делятся: ***d*** = НОД(***a,b***). Если ***d*** = НОД(***a,b***) = 1, то числа ***a*** и ***b*** являются *взаимно простыми*.

Определение НОД подсказывает следующий алгоритм его вычисления. Если числа ***a*** и ***b*** заданы, то найдем все положительные делители числа ***a*** и все положительные делители числа ***b***.

Выберем все числа, входящие в оба множества, и возьмем наибольшее из них. Это число и будет НОД. Эта процедура совсем проста, однако она неэффективна при больших значениях чисел ***a*** и ***b***.

НОД можно подсчитать более эффективным способом. Эвклид приводит его предложениях 1 и 2 книги VII своих “Элементов”. Алгоритм Эвклида действует следующим образом. Разделим *a* на *b* с остатком. Назавем этот остаток r_1. Если $r_1 \neq 0$, то разделим b на r_1 с остатком. Пусть r_2 - остаток второго деления. Аналогично, если $r_2 \neq 0$, то разделим r_1 на r_2 и получим новый остаток r_3.

Таким образом, ***i*** – ый цикл алгоритма состоит из одного деления с остатком, причем делимое равно остатку, полученному в (***i – 2***) цикле, а делитель – остатку, полученному в (***i - 1***) цикле. Цикл повторяется до тех пор, пока не получим нулевого остатка.

Наименьший ненулевой остаток является ***наибольшим общим делителем*** чисел ***a*** и ***b***.

Применим алгоритм Эвклида для вычисления НОД чисел 1234 и 54. Деление с остатком выглядит следующим образом:

$$1234 = 54 \cdot 22 + 46;$$
$$54 = 46 \cdot 1 + 8;$$
$$46 = 8 \cdot 5 + 6;$$
$$8 = 6 \cdot 1 + 2;$$
$$6 = 2 \cdot 3 + 0.$$

Последний ненулевой остаток равен 2, поэтому НОД(1234, 54) = 2.

Отметим, что неполные частные не принимают непосредственного участия в подсчете НОД.

Запишем алгоритм в виде последовательности действий.

Ввод: натуральные числа ***a*** и ***b***, $a \geq b$.

Вывод: ***D*** равно НОД(***a, b***).

Шаг 1. Присвоить ***A = a*** и ***R = B = b***.

Шаг 2. Заменить значение ***R*** остатком от деления a на ***b*** и перейти к шагу 3.

Шаг 3. Если ***R = 0, D = B*** (т.е. ***B*** = НОД(***a, b***)) и перейти к выходу, в противном случае перейти к шагу 4.

Шаг 4. Заменить значение ***A*** на значение ***B***, значение ***B*** на значение ***R*** и возвратится к шагу 2.

Таким образом, для вычисления НОД необходимо выполнить несколько действий деления с остатком.

Расширенный алгоритм Эвклида.

У алгоритма Эвклида есть еще один более мощный вариант – алгоритм Кнута, автора знаменитой книги “Исскуство программирования”, позволяющий одновременно с НОД получить параметры, необходимые для подбора ключей в алгоритме RSA.

Расширенный алгоритм Эвклида приводит к следующей теореме.

Теорема. Пусть ***d*** – наибольший общий делитель натуральных чисел ***a*** и ***b***. Тогда, существуют такие целые числа α и β, что $\alpha * a + \beta * b = d$.

Примечание: *Рассмотрение данного алгоритма рекомендуется выполнить самостоятельно. Подробное описание данного алгоритма приведено в книге С. Коутинхо “Введение в теорию чисел, алгоритм RSA”, Млсква, Посмаркет, 2001. (стр.54-58).*