ЛАБОРАТОРНАЯ РАБОТА № 7

ШИФРОВАНИЕ МЕТОДОМ СКОЛЬЗЯЩЕЙ ПЕРЕСТАНОВКИ

Цель работы: исследование шифра скользящей перестановки с использованием программной реализации XY-Mover.

Описание лабораторной работы. ***Устойчивые закономерности открытого текста и их использование при дешифровании шифров простой замены и перестановки.*** Возможность дешифрования какого-либо шифра в значительной мере зависит от того, в какой степени криптографические преобразования разрушают вероятностно-статистические закономерности, присутствующие в открытом содержательном тексте. Так, в осмысленных текстах любого естественного языка различные буквы встречаются с разной частотой, при этом относительные частоты букв в различных текстах одного языка близки между собой. То же самое можно сказать и о частотах пар, троек букв открытого текста. Кроме того, любой естественный язык обладает так называемой избыточностью, что позволяет с большой вероятностью «угадывать» смысл сообщения, даже если часть букв в сообщении не известна.

Таблицы относительных частот появления букв в тексте (табл. 2.3) приводятся в разных книгах. Они получены на основе подсчетов частот на больших объемах открытого текста. Учитывая, что для экспериментов берется различный исходный материал, значения вероятностей несколько отличаются между собой.

Таблица 2.3

1	а - 0,062	12	л - 0,035	23	ц - 0,004
2	б - 0,014	13	м - 0,026	24	ч - 0,012
3	в - 0,038	14	н - 0,053	25	ш - 0,006
4	г - 0,013	15	о - 0,090	26	щ - 0,003
5	д - 0,025	16	п - 0,023	27	ы - 0,016
6	е,е - 0,072	17	р - 0,040	28	ъ,ь - 0,014
7	ж - 0,077	18	с - 0,045	29	э - 0,003
8	з - 0,016	19	т - 0,053	30	ю - 0,006
9	и - 0,062	20	у - 0,021	31	я - 0,018
10	й - 0,010	21	ф - 0,002	32	- 0,175
11	к - 0,028	22	х - 0,009		

Если упорядочить буквы по убыванию вероятностей, то мы получим следующий вариационный ряд:

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

Например, в слове **СЕНОВАЛИТР** содержатся 10 наиболее часто встречающихся букв.

Частоты знаков алфавита зависят не только от языка, но и от характера текста. Так, в тексте по криптографии будет повышена вероятность букв «Ф», «Ш» из-за часто встречающихся слов «шифр», «криптография». В некоторых математических текстах может быть завышена частота буквы «Ф» из-за слов «функция», «функционал» и т.п. В стандартных текстовых файлах наиболее частым является символ «пробел». Частотная диаграмма содержательных текстов является устойчивой характеристикой текста. Из теории вероятностей следует, что при достаточно слабых ограничениях на вероятностные свойства случайного процесса справедлив закон больших чисел, т.е. относительные частоты $\frac{\vartheta_k}{N}$ знаков сходятся по вероятности к значениям их вероятностей p_k

$$P\left\{\left|\frac{\vartheta_k}{N} - p_k\right| > \varepsilon\right\} \xrightarrow[N\to\infty]{} 0.$$

Шифры перестановки и простой замены не полностью разрушают вероятностно-статистические свойства, имеющиеся в открытом сообщении.

При дешифровании текста, зашифрованного шифром простой замены, используют частотные характеристики открытого текста. Именно если подсчитать частоты встречаемости знаков в шифрованном тексте, упорядочить их по убыванию и сравнить с вариационным рядом вероятностей открытого текста, то эти две последовательности будут близки. Скорее всего на первом месте окажется пробел, далее будут следовать буквы «О», «Е», «А», «И».

Конечно, если текст не очень длинный, то не обязательно полное совпадение. Может оказаться на втором месте «О», а на третьем «Е», но в любом случае в первых и вторых рядах одинаковые буквы будут располагаться недалеко друг от друга, и чем ближе к началу (чем больше вероятность знаков), тем меньше будет расстояние между знаками.

Аналогичная картина наблюдается и для пар соседних букв, биграмм, открытого текста (наиболее частая биграмма русского открытого текста — СТ). Однако для получения устойчивой картины длина анализируемой последовательности должна быть достаточно большой. На сравнительно небольших отрезках открытого текста эта картина как-то смазана. Более устойчивой характеристикой биграмм является отсутствие в осмысленном тексте некоторых биграмм, как говорят, наличие запретных биграмм, имеющих вероятность, равную практически нулю.

Видели ли вы когда-нибудь в открытом тексте биграмму «ЪЬ» или биграммы вида «гласная» Ь, «пробел» Ь? Знание и использование указанных особенностей открытого текста значительно облегчает дешифрование шифра перестановки и замены.

Шифр перестановки. Положим X — множество открытых (содержательных) текстов в алфавите I. Длины всех возможных открытых текстов кратны величине T. Множеством ключей является симметрическая группа подстановок S_T степени T, для $g \in S_T$ определим функцию шифрования f_g, положив для $(i_1, i_2, ..., i_T) \in X$

$$f_g(i_1, i_2, ..., i_T) = (i_{g(1)}, i_{g(2)}, ..., i_{g(T)}),$$

доопределим f_g на остальных элементах из X по правилу: текст $x \in X$ делится на отрезки длины T и каждый отрезок длины T шифруется на ключе g по указанному выше закону шифрования. Последовательность, составленная из букв образов зашифрованных слов, является шифрованным текстом, соответствующим открытому тексту x и ключу g. Для шифрования текста длины, не кратной T, его дополняют буквами до длины, кратной T.

Дешифрование шифра перестановки. Шифрованный текст записывается в таблицу с T столбцами. Для восстановления открытого текста шифра перестановки нам необходимо переставить колонки таким образом, чтобы в строках появился осмысленный текст.

Рассмотрим пример дешифрования шифра перестановки восьми столбцов. Пусть шифротекст имеет следующий вид (табл. 2.4).

Таблица 2.4

1	2	3	4	5	6	7	8
п	а	я		в		и	м
о	ч	ш	г		у		е
е	б	ж	л		е		о
м		ч		о	т	о	я
	е	г	е		у	с	щ
	а	к	ь	з	а	т	т
я	р	е		е	п		ь
	ю	з	в	а	н	в	
о	й	а	в	е	ш	л	
	е	е	я	м		п	н
ь	р	р	н	з	е	е	е
з	а	м	а	н		а	к
ч	с	т	а		ь	а	н
о	я	л	м		а	л	

Окончание

1	2	3	4	5	6	7	8
о	ь	ч	х	т	а	т	
в		е	о	а	л	е	п
о	е	р	м	т	ь	е	
д	с	г	ы		о	а	т
е	б	в	н		ы		
	а	у	и	н	з	н	л
	г	и	а	о	к	к	д
	а	о	б	д	г	н	
	ж	а	у	е	д	я	д
х	л	и		е	м	о	а
к	р	т	д		ь	о	е
	ь	х	в	т	о	н	
р	л	е		е	д	а	ю
р		з	е	в		е	д
ш		в	а	е	н	е	н
т	и	й	е	в			д
		в		с	д		

Сопоставим перестановке столбцов таблицу 8×8, при этом поставим на пересечении i-й строки и j-й столбца единицу, если j-я колонка после обратной перестановки должна следовать за i-й. Наша задача — восстановить таблицу, отвечающую правильной перестановке столбцов.

Давайте теперь попарно пристраивать один столбец к другому. Если при этом в некоторых строках появляются запретные биграммы, то столбцы не могут в открытом тексте следовать друг за другом и соответствующая клеточка зачеркивается. В нашем примере шестой столбец не может следовать за четвертым, так как иначе в тексте в первой строке будет подряд два пробела. Посмотрим, например, шестую строку. Если бы четвертый столбец следовал за первым, то в тексте были бы слова, начинающиеся с «Ь».

После просмотра всех строк получим табл. 2.5.

Таблица 2.5

	1	2	3	4	5	6	7	8
1	x	x		x	x	x		x
2		x		x		x		
3			x					x
4	x	x		x		x	x	x

Окончание

	1	2	3	4	5	6	7	8
5	x				x	x	x	x
6	x	x		x		x	x	
7	x			x	x	x	x	x
8	x	x			x	x	x	x

Если бы текст был подлиннее и строк было бы побольше, то в каждой строке и в каждом столбце осталось бы ровно по одной не зачеркнутой клеточке и перестановка была бы восстановлена.

В таблице 2.5 только можно утверждать, что шестой столбец следует за третьим (обозначим это событие следующим образом: $3 \to 6$), если шестой столбец не является последним. Для шестого столбца может быть два варианта продолжения

$$\begin{array}{l} 8 \\ \uparrow \\ 3 \to 6 \to 5 \end{array}$$

Нам надо рассмотреть оба и постараться отсеять ложный вариант. Если отсеять ложный вариант не удастся, то надо продолжать оба варианта

$$\begin{array}{llll} & 8 & \to 4 & 1 \\ & \uparrow & & \uparrow \\ 3 \to & 6 \to 5 & \to 2 & \to 7 \end{array}$$

В итоге получаем некоторое дерево возможного следования столбцов в открытом тексте.

$$\begin{array}{l} \qquad\qquad\qquad\quad 7 \\ \qquad\qquad\qquad\quad \uparrow \\ 3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7 \\ \qquad\quad \downarrow \quad \downarrow \quad \downarrow \\ \qquad\quad 8 \quad 4 \quad 1 \to 7 \\ \qquad\quad \downarrow \quad \downarrow \\ \qquad\quad 4 \quad 1 \to 7 \\ \qquad\quad \downarrow \\ \qquad\quad 5 \\ \qquad\quad \downarrow \\ \qquad\quad 2 \to 7 \\ \qquad\quad \downarrow \\ \qquad\quad 1 \to 7 \end{array}$$

Каждой ветви дерева соответствует некоторая перестановка столбцов.

Далее проверяем каждый вариант на осмысленность и получаем правильный вариант

$$3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7$$

Заметим, что не обязательно было строить дерево до конца. Например, ветвь $3 \to 6 \to 8 \to 4 \to 5$ можно было отсеять сразу. Разве можно признать осмысленным фрагмент текста, приведенный в табл. 2.6?

Таблица 2.6

3	6	8	4	5
я		м		в
ш	у	е	г	
ж	е	о	л	
ч	т	я		о
г	у	щ	е	з
к	а	т	ь	е
е	а	т		а

Такая процедура отсечения ветвей была бы просто необходима, если бы строк было поменьше и дерево было бы соответственно гораздо ветвистей. Предложенную процедуру легко автоматизировать и сделать пригодной для реализации на ЭВМ. Алгоритм дешифрования должен состоять из следующих этапов.

1. Предварительная работа. Анализируя достаточно представительный объем открытых текстов, построить множество запретных биграмм.

2. Предварительная работа. Составить словарь всех возможных v-грамм для $v = 2, 3, \ldots, d$, которые могут встретиться в открытом тексте. Число d выбирается исходя из возможностей вычислительной техники.

Построить таблицу 8×8. При этом перебираются последовательно все запретные биграммы и для каждой опробуемой биграммы — последовательно все строки. Если хотя бы в одной строке первый символ биграммы встречается в i-м столбце, а второй — в j-м, то клеточка $i \times j$ таблицы зачеркивается.

3. Выбрать некоторый столбец в качестве начального.

4. Начать процедуру построения дерева путем пристраивания к исходному столбцу всех вариантов столбцов.

5. Для каждого полученного варианта добавить еще один из оставшихся столбцов. Если хотя бы в одной из строк таблицы встретится 3-грамма, которая отсутствует в словаре размещенных 3-грамм, то вариант отсеивается.

6. Для каждого из неотсеянных вариантов добавляем еще один столбец и проводим отсев ложных вариантов по словарю разрешенных 4-грамм.

Если словарь был построен только для $d \le 3$, то отсев проводится путем проверки на допустимость 3-грамм, встретившихся в последних трех столбцах каждой строки. Продолжаем этот процесс до получения полной перестановки.

В таблице 2.7 приведен восстановленный для нашего примера текст.

Таблица 2.7

	1	2	3	4	5	6	7	8
1	я		в	а	м		п	и
2	ш	у		ч	е	г	о	
3	ж	е		б	о	л	е	
4	ч	т	о		я		м	о
5	г	у		е	щ	е		с
6	к	а	з	а	т	ь		т
7	е	п	е	р	ь		я	
8	з	н	а	ю		в		в
9	а	ш	е	й		в	о	л
10	е		м	е	н	я		п
11	р	е	з	р	е	н	ь	е
12	м		н	а	к	а	з	а
13	т	ь		с	н	а	ч	а
14	л	а		я		м	о	л
15	ч	а	т	ь		х	о	т
16	е	л	а		п	о	в	е
17	р	ь	т	е		м	о	е
18	г	о		с	т	ы	д	а
19	в	ы		б		н	е	
20	у	з	н	а	л	и		н
21	е	к	о	г	д	а		к
22	о	г	д	а		б		н

Окончание

	1	2	3	4	5	6	7	8
23	а	д	е	ж	д	у		я
24	и	м	е	л	а		х	о
25	т	ь		р	е	д	к	о
26	х	о	т	ь		в		н
27	е	д	е	л	ю		р	а
28	з		в		д	е	р	е
29	в	н	е		н	а	ш	е
30	й		в	и	д	е	т	ь
31	в	а	с					

Дальнейшее развитие шифры перестановки получили осуществлением идеи непрерывной локальной перестановки символов открытого текста под действием управляющей последовательности. Для осуществления перемешивания знаков открытого текста в памяти шифратора запоминаются отдельные знаки текста и проводится задержка их передачи в дискретном времени. Введем параметры n_1 и n_2 так, что $n = n_1 + n_2$. В этих шифрах i-й знак передаваемого сообщения переставляется в шифрованном сообщении на j-е место, где $i - n_1 \le j \le i + n_2$.

Управляющую последовательность временем задержки стараются выбрать так, чтобы время задержки каждого символа было случайной величиной с равномерным распределением, т.е. вероятность каждого фиксированного значения времени задержки должна быть близка к $1/n$.

Шифрующий автомат скользящей перестановки. Рассмотрим схему шифрующего автомата, позволяющего при подходящей управляющей последовательности реализовать произвольный шифр скользящей перестановки (рис. 2.5).

На вход узла формирования задержки (УФЗ) в каждом такте t подается вектор $\vec{y} = \left(y_1^t, \dots y_n^t\right)$,

$$y_i^t \in \{0,1\},\ i = \overline{2, n-1},\ y_1^t = y_n^t = 1.$$

Узел формирования задержки является конечным автоматом $\left(F_2^n, F_2^n, \{1, \dots, n\}, \delta, \lambda\right)$, множеством состояний которого являетсямножество всевозможных двоичных векторов — заполнений $\vec{x}(t) = (x_1^t, \dots, x_n^t)$ нижнего накопителя. В такте t вырабатывается натуральное число — значение γ_t задержки

$$\gamma_t = \max j : \left\{x_j^t = y_j^t = 1, j = \overline{1, n}\right\}.$$

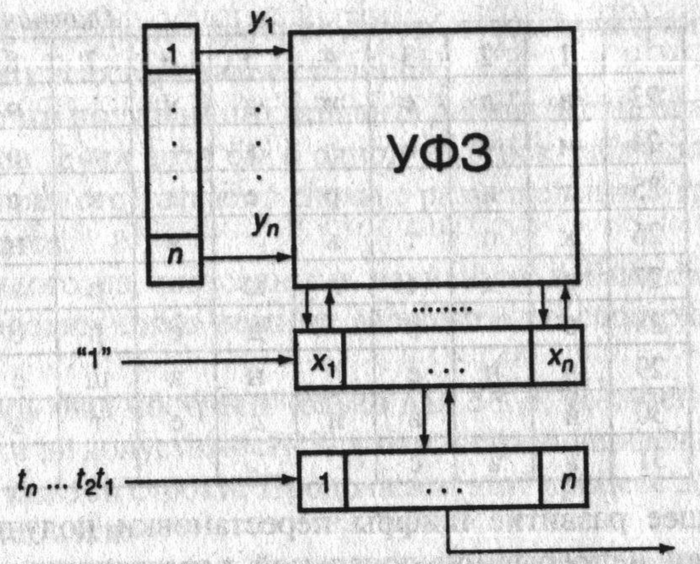

Рис. 2.5. Схема реализации шифрующего автомата скользящей перестановки

При этом автомат переходит в следующее состояние:

$$\bar{x} = (t+1) = (x_1^{t+1}, \dots, x_n^{t+1}),$$

где

$$x_1^{t+1} = 1,$$

$$x_i^{t+1} = \begin{cases} 0, i = \gamma_t + 1 \\ x_{i-1}^t, i \in \{2, \dots, \gamma_t, \gamma_t + 2, \dots, n\}. \end{cases}$$

Знаки открытого текста записываются на нижний накопитель. В линию в t-м такте посылается знак открытого текста, записанный в ячейке с номером γ_t. Состояния автомата $\vec{x}(t)$ являются индикаторами, показывающими, какие из знаков открытого текста еще не считаны с нижнего накопителя.

Для зашифрования последовательности $t_N, \dots, t_2, t_1$ поступают следующим образом. Сначала записываем в нижний накопитель первые n_1 знаков открытого текста

$$(t_{n1}, \dots, t_1, 0, \dots, 0).$$

Одновременно автомат устанавливается в начальное состояние

$$\vec{x}(1) = (0, \dots, 0).$$

После этого автомат УФЗ начинает работать по описанному выше закону до тех пор, пока в накопитель не поступит последний

знак t_N открытого текста. С прекращением подачи на накопитель знаков открытого текста происходит прекращение подачи единиц на накопитель-индикатор. В оставшиеся n_1 тактов производится считывание записанной в накопителе информации.

При расшифровывании УФЗ работает по той же схеме, только вместо считывания необходимо организовывать запись информации во второй накопитель.

Рассмотрим особенности работы УФЗ. В каждом такте t (за исключением последних n_1 тактов) в накопителе-индикаторе $\vec{x}(t)$ записано ровно n_1 единиц. Поэтому в такте t величина задержки может принимать только одно из n_1 значений в интервале $\{1, \dots, n\}$. В частном случае, когда $n = 1$ либо $n_1 = n$, УФЗ вырабатывает постоянно значения задержки $\gamma_t = 1$ и $\gamma_t = n$ соответственно. Легко видеть, что результирующее преобразование открытого текста действительно будет шифром скользящей перестановки. Условие $y_1^t = 1, t = 1, 2, \dots$ обеспечивает постоянное считывание во всех тактах, а условие $y_2^t = 1$ ограничивает величину задержки $\gamma_t \le n$.

Пример 2.2

Примем $n = 7$, $n_1 = 3$, $n_2 = 4$. На вход УФЗ на каждом шаге t работы шифрующего автомата подается вектор $\vec{y} = (y_1^t, \dots, y_n^t)$, получаемый в линейном регистре сдвига (ЛРС) (рис. 2.6).

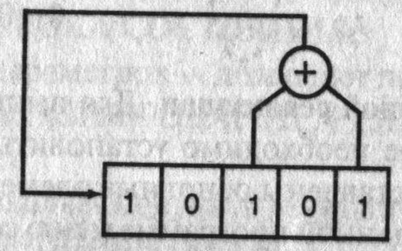

Рис. 2.6. Линейный регистр сдвига

Надеемся, что читатель сможет написать последовательность состояний данного линейного регистра сдвига, с помощью которой образуется управляющая последовательность шифрующего автомата.

Будем обозначать на каждом шаге работы шифрующего автомата последовательность $y_1, y_2, \dots, y_n$ (в нашем случае $y_1, y_2, \dots, y_7$), поступающую с ЛРС, как «Y», а последовательность единиц $x_1, x_2, \dots, x_n$ (в нашем случае $x_1, x_2, \dots, x_7$) как «1». В нижней строке будем записывать знаки открытого текста, находящиеся на данном шаге в нижнем накопителе шифрующего автомата $t_1, t_2, \dots, t_n$ (в нашем случае $t_1, t_2, \dots, t_7$). Рассмотрим пошагово работу шифратора при конкретных условиях.

На каждом шаге, начиная с левого края и идя направо, мы искали первое совпадение в строках Y и 1 (1 в обеих строках) и для удобства обводили этот столбец. Элемент открытого текста, который оказался в выбранном столбце, уходит в линию. Таким образом, в нашем примере последовательность, ушедшая в линию, имеет следующий вид:

$$t_2, t_4, t_5, t_6, t_1, t_7, t_3, t_8.$$

1-й шаг

"Y"	1	1	0	1	0	1	1
"1"	1	1	1	0	0	0	0
	t_3	t_2	t_1	0	0	0	0

2-й шаг

"Y"	1	0	1	0	1	0	1
"1"	1	1	0	1	0	0	0
	t_4	t_3	t_2	t_1	0	0	0

3-й шаг

"Y"	1	0	0	1	0	1	1
"1"	1	0	1	0	1	0	0
	t_5	t_4	t_3	t_2	t_1	0	0

4-й шаг

"Y"	1	0	0	0	1	0	1
"1"	1	0	0	1	0	1	0
	t_6	t_5	t_4	t_3	t_2	t_1	0

5-й шаг

"Y"	1	0	0	0	0	1	1
"1"	1	0	0	0	1	0	1
	t_7	t_6	t_5	t_4	t_3	t_2	t_1

6-й шаг

"Y"	1	1	0	0	0	0	1
"1"	1	1	0	0	0	1	0
	t_8	t_7	t_6	t_5	t_4	t_3	t_2

7-й шаг

"Y"	1	0	1	0	0	0	1
"1"	0	1	0	0	0	0	1
	0	t_8	t_7	t_6	t_5	t_4	t_3

8-й шаг

"Y"	1	0	0	1	0	0	1
"1"	0	0	1	0	0	0	0
	0	0	t_8	t_7	t_6	t_5	t_4

Описание программной реализации. Для выполнения лабораторной работы на компьютере необходимо установить программный модуль XY-Mover. Ниже представлены основные элементы программы.

1. Строка меню. В данной программе меню состоит из трех пунктов: ШИФРОВАНИЕ, ВИД, ПОМОЩЬ. Необходимый пункт меню можно выбрать, щелкнув по нему левой кнопкой мыши, или с помощью кнопок клавиатуры «вправо», «влево», нажав перед этим функциональную клавишу F10. После того как пользователь выбрал необходимый ему пункт меню, откроется ниспадающее подменю. Рассмотрим пункты меню подробнее.

■ ШИФРОВАНИЕ — пункты ниспадающего меню можно выбрать либо левой кнопкой мыши, либо кнопками клавиатуры «↑», «↓». Рассмотрим подробнее пункты подменю:

— РЕДАКТИРОВАНИЕ ПАРАМЕТРОВ — позволяет задать необходимые параметры в поле окна программы «Параметры шифратора»,

— ШИФРОВАТЬ — запускает процесс шифрования,

— ДЕШИФРОВАНИЕ — запускает процесс дешифрования,

— ВЫХОД — завершение работы программы;

■ ВИД — этот пункт меню позволяет выбрать внешний вид программы. Ниже приводятся пункты подменю:

— ПАРАМЕТРЫ — позволяет использовать поле «Параметры шифратора», описанную ниже;

— СХЕМА ШИФРАТОРА — позволяет наблюдать структурную схему шифрующего автомата;

— СТРОКА СОСТОЯНИЯ — выбор этого пункта меню позволяет наблюдать строку состояния, описанную ниже.

2. Панель инструментов. Возможно, пользователю будет удобнее воспользоваться панелью инструментов (так называемыми кнопками) вместо работы с меню. Кнопки дублируют некоторые пункты меню, но выбрать кнопку гораздо удобнее, чем пункт подменю. Для этого необходимо щелкнуть по выбранной кнопке левой кнопкой мыши. Рассмотрим кнопки слева направо:

■ А, позволяет получить результат, аналогичный пункту меню: ВИД/ПАРАМЕТРЫ;

■ схема шифратора — позволяет получить результат, аналогичный пункту меню ВИД/СХЕМА шифратора;

■ строка состояния — позволяет получить результат, аналогичный пункту меню: ВИД/СТРОКА СОСТОЯНИЯ;

■ редактирование параметров — позволяет получить результат, аналогичный пункту меню ШИФРОВАНИЕ/РЕДАКТИРОВАНИЕ ПАРАМЕТРОВ;

■ шифровать — позволяет получить результат, аналогичный пункту меню ШИФРОВАНИЕ/ШИФРОВАТЬ;

■ «дешифровать» — позволяет получить результат, аналогичный пункту меню ШИФРОВАНИЕ/ДЕШИФРОВАТЬ;

■ «прервать» — позволяет прервать процесс обработки данных (шифрование или дешифрование).

3. Строка состояния. Строка состояния находится в нижней части окна программы. На ней выводится информация о состоянии программы:

■ шифрование методом скользящей перестановки — это сообщение указывает на то, что программа готова к работе.

■ завершено ... % — индикация объема выполненной работы при зашифровании или расшифровании.

4. Описание полей окна программы:

■ параметры шифратора:

— $n1$ — число знаков, которые записываются в нижний накопитель первыми,
— $n2$ — остальные знаки (всего их 127),
— отводы регистра — точки съема ЛРС,
— начальное заполнение — начальное заполнение ЛРС, которое пользователь может изменять;

- входной поток:
  — поле, в котором отражается имя текстового файла, содержание которого нужно шифровать/расшифровать,
  — кнопка «Открыть» — при нажатии на эту кнопку открывается стандартное для Windows окно «ОТКРЫТИЕ ФАЙЛА»,
  — текст — содержимое файла, открытого для шифрования/расшифрования,
  — битовый поток — побитное представление символов выбранного файла;
- выходной поток:
  — поле, в котором отражается имя текстового файла, содержание которого нужно расшифровать/шифровать,
  — кнопка «Открыть» — при нажатии на эту кнопку открывается стандартное для Windows окно «ОТКРЫТИЕ ФАЙЛА»,
  — текст — содержимое файла, открытого для расшифрования/шифрования,
  — битовый поток — побитное представление символов выбранного файла.

Задание

1. Для выполнения лабораторной работы на компьютере необходимо установить программный модуль XY-Mover.
2. Выполнить начальные установки шифратора согласно примеру.
3. Загрузить файл для шифрования.
4. Произвести шифрование информации с использованием шифра скользящей перестановки, сохранить шифртекст в файле.
5. Описать в отчете процесс шифрования и расшифрования данных с использованием программы-эмулятора XY-Mover. Проанализировать полученные данные.
6. Включить в отчет о лабораторной работе ответы на контрольные вопросы, выбранные в соответствии с номером варианта (табл. 2.8).

Таблица 2.8

Номер варианта	Контрольные вопросы
1, 5, 7, 3, 9, 18, 28	Почему шифрование методом гаммирования является наиболее подходящим для высокоскоростных линий телекоммуникационной связи?
2, 4, 6, 8, 20, 22, 24, 26, 30	Какие общие требования предъявляются к гамме шифра?
11, 13, 15, 10, 17, 19, 27	Приведите пример, поясняющий работу шифрующего автомата скользящей перестановки при $n = 5$, $n_1 = 2$, $n_2 = 3$
12, 14, 16 21, 23, 25, 29	Кратко опишите работу схемы реализации шифра скользящей перестановки