УДК 681.3.07

д.т.н., проф. Белецкий А.Я., Аксентий Е.А.
(Национальный авиационный университет)

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ КРИПТОГРАФИЧЕСКИХ ПРИМИТИВОВ ТИПА ПЕРЕСТАНОВКИ ЭЛЕМЕНТОВ ШИФРУЕМОГО БЛОКА

Введение и постановка задачи

Как известно [1], криптостойкие системы могут быть построены на основе многократного (итерационного) применения относительно простых криптографических преобразований (примитивов). Родоначальник компьютерной криптографии Клод Шеннон еще в середине пятидесятых годов прошлого столетия предложил алгоритм зашифрования открытых текстов, содержащий всего лишь два криптографических примитива [2]. Один из примитивов осуществляет операцию нелинейной подстановки элементов шифруемого текста (Substitution). Вторым примитивом (по Шеннону) является оператор перестановки (Permutation). Указанная пара примитивов (Substitution + Permutation) совместно образуют один раунд криптографического преобразования так называемой SP-сети [3]. В современных шифраторах используются и другие криптографические примитивы. К числу таких примитивов относятся: гаммирование, циклический сдвиг, скользящее кодирование и ряд других [4].

В данной статье разрабатывается универсальный программный моделирующий комплекс, реализующий перестановку элементов шифруемого блока. Элементом перестановки может быть один, два, четыре, восемь, 16 или 32 бита, в зависимости от длины N шифруемого блока и степени n неприводимого полинома φ_n, участвующих в операции перестановки (табл. 1).

Таблица 1. Элементы перестановки блоков шифруемых текстов

Степень полинома (n)	$N = 64$	$N = 128$	$N = 256$	$N = 512$	Размерность S - боксов
8	–	–	Бит	2 бита	16x16
7	–	Бит	2 бита	Полубайт	16x8
6	Бит	2 бита	Полубайт	Байт	8x8
5	2 бита	Полубайт	Байт	Слово	8x4
4	Полубайт	Байт	Слово	Дв. слово	4x4

Алгоритм перестановки задается преобразованием

$$y = (x \oplus \alpha)^{-1}_{\varphi_n} \otimes A \oplus \beta, \qquad (1)$$

согласно которому элемент, находящийся в ячейке блока по адресу x, перемещается в ячейку этого же блока по адресу y.

Таким образом, перестановка элементов шифруемого блока осуществляется с помощью обобщенной нелинейной подстановки (S-боксов), заданной соотношением (1). В преобразовании (1) приняты такие обозначения: $\oplus$ и $\otimes$ - операторы поразрядного суммирования и матричного умножения в кольце вычетов по mod 2 соответственно; α и β - аддитивные компоненты (двоичные векторы); A - квадратная невырожденная (0, 1)-

матрица преобразования; $(a)_{\varphi_n}^{-1}$ - обратный по умножению над полиномом φ_n двоичный вектор a. Порядок матрицы и всех двоичных векторов преобразования совпадает со степенью n неприводимого полинома φ.

Программно-моделирующий комплекс включает ряд интерфейсов, краткая характеристика которых приводится ниже.

Базовый интерфейс программного комплекса. С помощью данного интерфейса (рис. 1) осуществляется параметризация универсального блока перестановки, который для краткости будем именовать в дальнейшем как UBP (Universal Block Permutation) модуль, а точнее – модуль параметризации UBP.

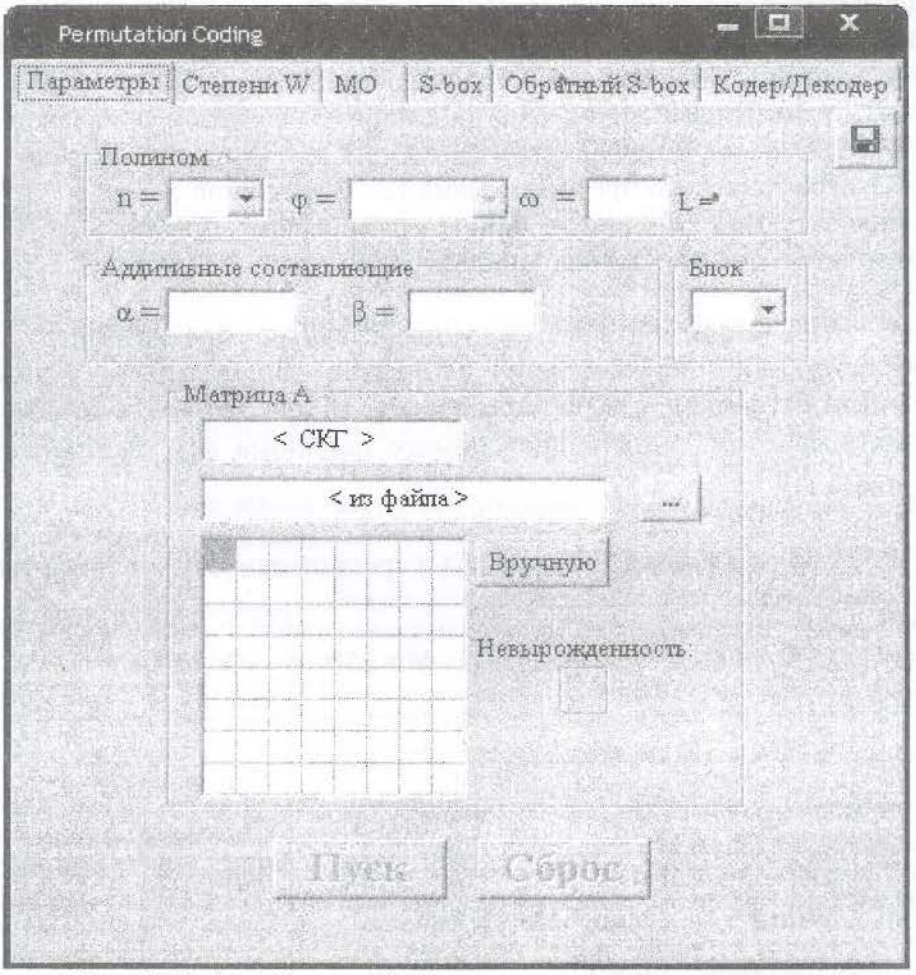


Рис. 1. Базовый интерфейс моделирующего комплекса

Интерфейс содержит верхнюю линейку информационных окон, в которую входят собственно окно «Параметры», а также окна «Степени W», мультипликативно-обратных значений «MO», «S-box», «Обратный S-box» и окно «Кодер/Декодер».

Ниже линейки информационных окон располагается линейка окон, с помощью которых конкретизируются параметры неприводимых полиномов. В состав этих параметров входят степень полинома n, векторное значение полинома φ и образующий элемент ω поля Галуа $GF(2^n)$. Если параметры n, φ и ω установлены, то, нажав на кнопку ПУСК, можно вывести значение L, равное показателю полинома φ_n относительно элемента ω.

Далее под линейкой окон параметров неприводимых полиномов (НП) размещена линейка окон, посредством которых устанавливаются значения двоичных аддитивных компонент α и β, а также размер шифруемого блока.

И, наконец, последнюю группу составляют окна, выделенные для размещения матрицы преобразования, обозначенную на интерфейсе, как матрица $\grave{A}$. Порядок матрицы, как отмечалось выше, совпадает со степенью n неприводимого полинома φ. В качестве $\grave{A}$ может быть выбрана матрица, отвечающая произвольному *составному коду Грея* (СКГ) [5]. С этой целью в окно <СКГ> необходимо ввести цифровой эквивалент, соответствующий тому или иному коду. Отличительная особенность СКГ состоит в том, что отвечающие им матрицы преобразования являются гарантированно невырожденными, т.е. их определитель по mod 2 равен единице. Матрицы $\grave{A}$ также могут быть введены или <из файла>, размещенного в той или иной папке на любом диске компьютера, или <Вручную>. Для проверки невырожденности следует нажать на кнопку ПУСК. Если введенная матрица невырождена, то в соответствующем окне появится зеленого цвета знак «плюс», в противном случае – красный знак «минус».

На рис. 2 показан пример параметризации базового модуля UBP.

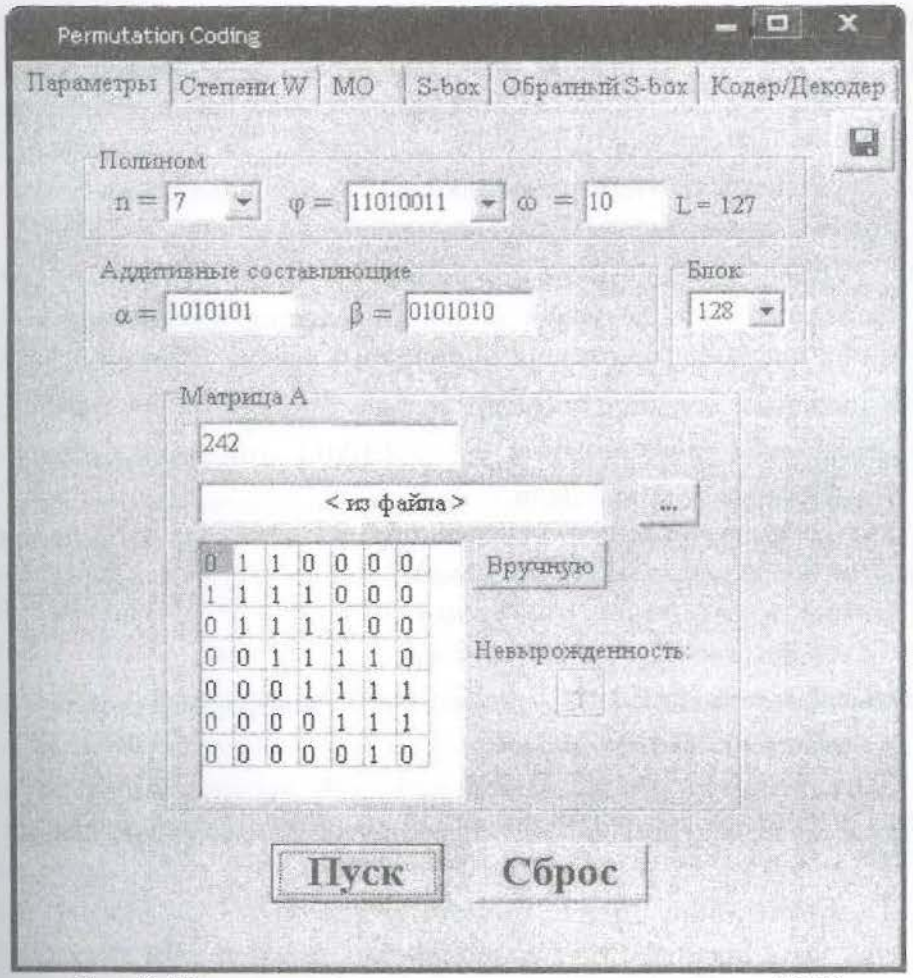


Рис. 2. Параметризация перестановочного модуля UBP

В соответствии с приведенными на рис. 2 параметрами криптографического примитива (длина блока 128 бит, неприводимый полином седьмой степени) по табл. 1 находим, что элементом перестановки шифруемых блоков будет один бит. Обратим внимание на то, что окна, с помощью которых устанавливается Степень полинома, выбираются Неприводимый полином и длина Блока, снабжены «прокрутками».

Степени примитивных элементов поля $GF(2^n)$. Полное множество ненулевых элементов поля $GF(2^n)$ над неприводимым полиномом φ может быть представлено в виде

степеней ω^k *примитивного* (образующего) *элемента* ω, вычисляемых по mod φ. Таким образом, примитивным элементом является такой $m-$разрядный, $2 \le m < n$, двоичный вектор ω, степени которого по mod φ образуют *последовательность максимальной длины*

$$L = 2^n - 1, \tag{1}$$

т.е., так называемую $M-$*последовательность*. Это означает, в частности, что ненулевые компоненты $GF(2^n)$, сформированные степенями примитивного элемента ω по mod φ, составляют циклическую абелеву группу относительно операции умножения, причем

$$(\omega^L)_\varphi = 1. \tag{2}$$

Проанализируем соотношения (1) и (2). Выражение (1) определяет число ненулевых элементов поля $GF(2^n)$. А теперь предположим, что равенство (2) достигается при некотором значении степени $K < L$. Это будет означать, что, во-первых, на полном интервале (от 0 до $2^n - 1$) допустимых значений степеней k элемента ω укладываются несколько повторяющихся с периодом K последовательностей двоичных векторов. И, во-вторых, как следствие периодичности, степени ω по mod φ не обеспечивают формирования полного множества ненулевых элементов поля $GF(2^n)$. Следовательно, такой элемент ω не является образующим (примитивным) элементом данного поля.

Если для некоторого неприводимого полинома φ_n над $GF(2^n)$ примитивным является *минимальный* элемент $\omega = 10$, то такой полином является *примитивным*. Более строгое математическое определение примитивности полинома таково. Многочлен $\varphi_n(x)$ степени n называется примитивным, если он не делит нацело ни один многочлен вида $x^S - 1$, где $S < L = 2^n - 1$. Выбранный ранее полином $\varphi_7 = 11010011$ относится к классу *примитивных неприводимых полиномов* седьмой степени.

Поясним методику вычисления степеней образующего элемента $\omega = 10$ по модулю неприводимого многочлена $\varphi_7 = 11010011$. Имеем

$$\omega^0 = 1,$$
$$\omega^1 = 10.$$

Последующие $k-$е степени, $k \ge 2$, элемента ω будем формировать по правилу

$$\omega^k = \omega^{k-1} \cdot \omega^1. \tag{3}$$

Вычисления в соотношении (3) для $\omega = 10$ сводятся к сдвигу множимого ω^{k-1} на один разряд влево. Следуя данному алгоритму, получим

$$\omega^2 = 100;$$
$$\omega^3 = 1000;$$
$$\omega^4 = 10000;$$
$$\omega^5 = 100000;$$
$$\omega^6 = 1000000.$$

Седьмая степень (ω^7) примитивного образующего элемента $\omega = 10$, равная 10000000, оказывается восьмиразрядной. Следовательно, двоичный вектор (ω^7) должен быть приведен к остатку по модулю φ, являющимся полиномом седьмой степени, двоичный эквивалент которого представляет собой восьмиразрядный вектор. В двоичной модулярной арифметике операция поразрядного вычитания, появляющаяся на этапе вычисления остатков, эквивалентна операции поразрядного сложения. Таким образом,

$$(\omega^7) \bmod \varphi = (\omega^7) \oplus \varphi = \begin{array}{r} 10000000 \\ \oplus \\ \underline{11010011} \\ 1010011 = \omega^7. \end{array} \quad (4)$$

Продолжая вычисления (3) степеней элемента ω по модулю выбранного НП φ, приходим (определяя, в случае необходимости, остатки по формуле (4)) к результатам, которые представлены на рис. 3.

Permutation Coding

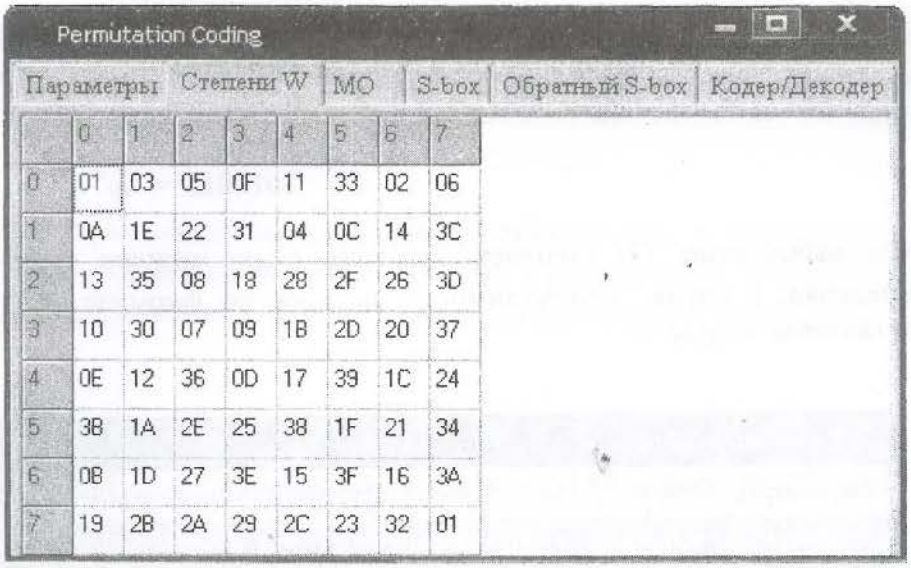
Параметры | Степени W | MO | S-box | Обратный S-box | Кодер/Декодер

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	02	04	08	10	20	40	53	75	39	72	37	6E	0F	1E	3C
1	78	23	46	5F	6D	09	12	24	48	43	55	79	21	42	57	7D
2	29	52	77	3D	7A	27	4E	4F	4D	49	41	51	71	31	62	17
3	2E	5C	6B	05	0A	14	28	50	73	35	6A	07	0E	1C	38	70
4	33	66	1F	3E	7C	2B	56	7F	2D	5A	67	1D	3A	74	3B	76
5	3F	7E	2F	5E	6F	0D	1A	34	68	03	06	0C	18	30	60	13
6	26	4C	4B	45	59	61	11	22	44	5B	65	19	32	64	1B	36
7	6C	0B	16	2C	58	63	15	2A	54	7B	25	4A	47	5D	69	01

Рис.3. Таблица степеней $\omega = 10$ по модулю $\varphi_7 = 11010011$

Таблица, показанная на рис. 3, выводится на экран монитора при нажатии на клавишу СТЕПЕНИ W. Показатель степени k примитивного образующего элемента $\omega = 10$ поля $GF(2^7)$ образован конкатенацией старшей компоненты k_2, эквивалентной трехразрядному двоичному вектору, и младшей компоненты k_1, являющейся эквивалентом двоичного полубайта. Таким образом, $k = k_2 \circ k_1$, где $\circ$ – знак конкатенации, причем k_2 - семеричные цифры, отложены на оси ординат, а k_1 - 16-ричные цифры, расположены на оси абсцисс таблицы. На пересечении абсциссы и ординаты таблицы на рис. 3 как раз и размещено значение k–й степени ω по $\bmod \varphi$.

Аналогичным образом вычисляются степени любого примитивного элемента ω для выбранного неприводимого полинома φ_n поля $GF(2^n)$. Для примера на рис. 4 показана таблица степеней образующего элемента $\omega = 11$ поля $GF(2^6)$ над неприводимым полиномом $\varphi_n = 1010111$. Показатель данного полинома φ_n (равный наименьшему положительному числу e, при котором $x^e - 1$ делится на полином φ_n) равен 21. Поскольку e оказалось меньше числа ненулевых элементов поля $GF(2^6)$, то полином $\varphi_n = 1010111$ не является примитивным. И, тем не менее, выбрав в качестве образующего элемент $\omega = 11$, мы имеем возможность сформировать полную абелеву циклическую группу, содержащую все 63 ненулевых элемента поля $GF(2^6)$.



	0	1	2	3	4	5	6	7
0	01	03	05	0F	11	33	02	06
1	0A	1E	22	31	04	0C	14	3C
2	13	35	08	18	28	2F	26	3D
3	10	30	07	09	1B	2D	20	37
4	0E	12	36	0D	17	39	1C	24
5	3B	1A	2E	25	38	1F	21	34
6	0B	1D	27	3E	15	3F	16	3A
7	19	2B	2A	29	2C	23	32	01

Мультипликативно обратные элементы поля $GF(2^n)$. Соотношения (1) и (2) приводят к достаточно простой схеме вычисления *мультипликативно обратных* (МО) или – обратных (инверсных) по умножению величин. В самом деле, согласно (3), для $n-$битных элементов, образующих поле $GF(2^n)$,

$$(\omega^k \cdot \omega^l) \bmod \varphi_n = 1, \qquad \text{если только } k+l = 2^n - 1. \tag{5}$$

Следовательно, при соблюдении условия (5) степени ω^k и ω^l являются взаимно обратными над неприводимым полиномом φ_n. На основании равенства (5) легко составить алгоритм формирования $n-$битных величин x^{-1}, инверсных двоичным векторам $n-$го порядка. Данный алгоритм включает следующие этапы преобразований x.

Этап 1. Рассчитываем степени образующего элемента ω (начиная с 0 до 2^n-1) по модулю неприводимого полинома φ_n и выписываем эти степени ($n-$разрядные двоичные числа) в колонку сверху вниз в порядке возрастания степени k элемента ω.

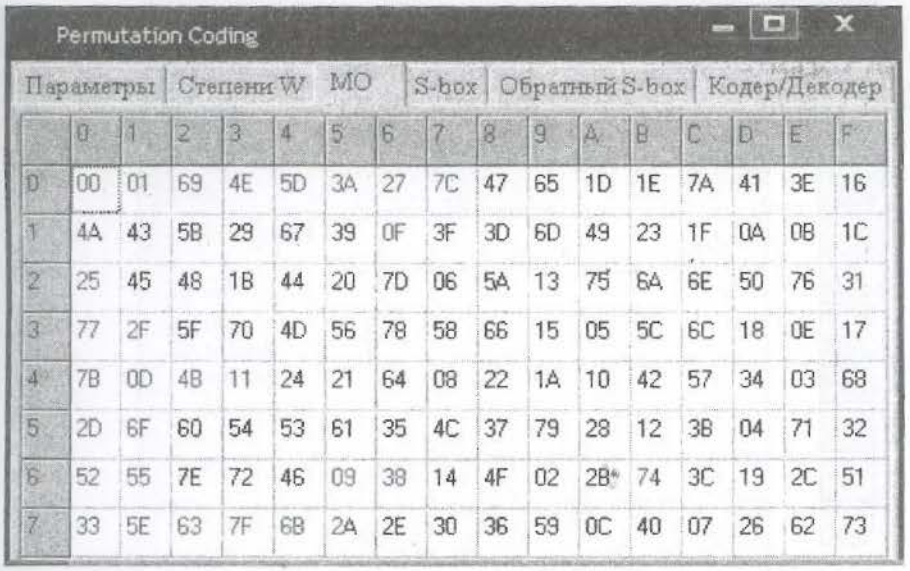
Этап 2. Размещаем справа относительно первой колонки вторую, образованную инверсией колонки степеней. В результате такой операции последний элемент (двоичный вектор) первой колонки окажется в верхней (первой) строке второй колонки. Предпоследний (второй снизу) элемент первой колонки будет размещен во второй сверху строке правой колонки и т.д.

Следовательно, после выполнения второго этапа преобразования в каждой k-й строке соседних колонок будут находиться взаимно обратные по mod φ_n двоичные $n-$разрядные числа.

Этап 3. Ранжируем строки левой колонки (и дружно с ними смежные элементы правой колонки), располагая их сверху вниз в порядке возрастания значений двоичных чисел.

В результате такой ранжировки левая колонка будет состоять из натуральной последовательности $n-$разрядных двоичных векторов x, а правая колонка – содержать значения МО величин x^{-1}, отвечающих числам x.

Наиболее удобно представлять множество мультипликативно обратных значений $\{x^{-1}\}$ не в виде колонки, а в форме таблицы, пример которой для НП $\varphi_7 = 11010011$ показан на рис. 4.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	69	4E	5D	3A	27	7C	47	65	1D	1E	7A	41	3E	16
1	4A	43	5B	29	67	39	0F	3F	3D	6D	49	23	1F	0A	0B	1C
2	25	45	48	1B	44	20	7D	06	5A	13	75	6A	6E	50	76	31
3	77	2F	5F	70	4D	56	78	58	66	15	05	5C	6C	18	0E	17
4	7B	0D	4B	11	24	21	64	08	22	1A	10	42	57	34	03	68
5	2D	6F	60	54	53	61	35	4C	37	79	28	12	3B	04	71	32
6	52	55	7E	72	46	09	38	14	4F	02	2B	74	3C	19	2C	51
7	33	5E	63	7F	6B	2A	2E	30	36	59	0C	40	07	26	62	73

Рис.4. Таблица МО по mod $\varphi_7 = 11010011$ величин

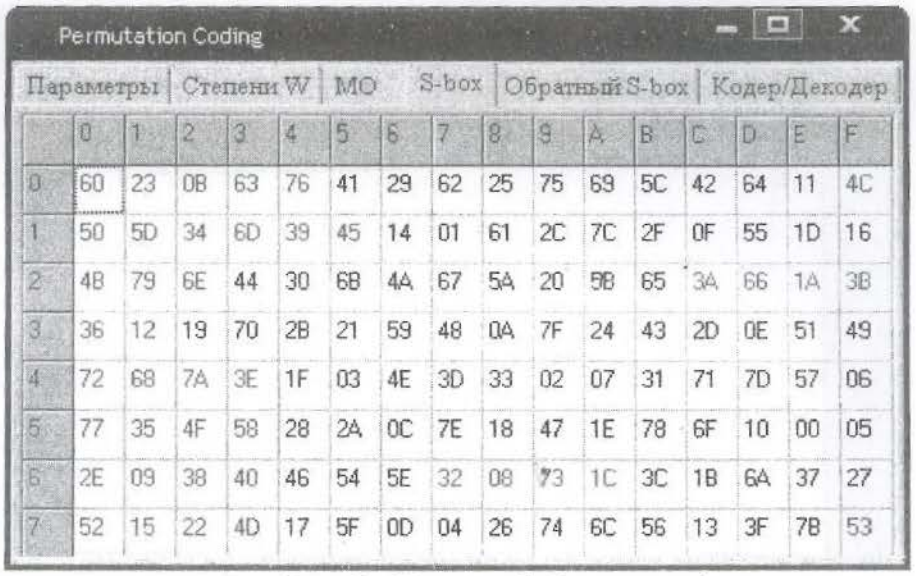
Таблицы инверсных по умножению чисел выводятся на панель результатов при нажатии на кнопку <MO> базового интерфейса (рис. 2). Координаты таблицы и значения МО представлены в 16-ричной системе счисления.

Функции S-бокса и обратного S-бокса. Посредством функции S-бокса как раз и реализуется преобразование (криптографический примитив), заданное соотношением (1). Этим преобразованием осуществляется нелинейная замена (подстановка) байта $\tilde{o}$ на байт $\acute{o}$ и, тем самым, вычисляется адрес $\acute{o}$, в который перемещается байт, находившийся по адресу $\tilde{o}$.

Как следует из выражения (1), данная функция предполагает выполнение ряда последовательных операций. Сначала адрес $\tilde{o}$ поразрядно складывается по mod 2 с двоичным вектором α, образуя байт $x' = x \oplus \alpha$. Затем по таблице, представленной на рис. 4, вычисляется мультипликативно обратное вектора $\tilde{o}'$ над НП $\varphi_7 = 11010011$. После этого МО $\tilde{o}'$ умножается в кольце вычетов по mod 2 на матрицу $\grave{A}$, представленную в поле матрицы на рис. 2. И, наконец, результат перемножения поразрядно суммируется по mod 2 с аддитивной компонентой β, образуя адрес $\acute{o}$.

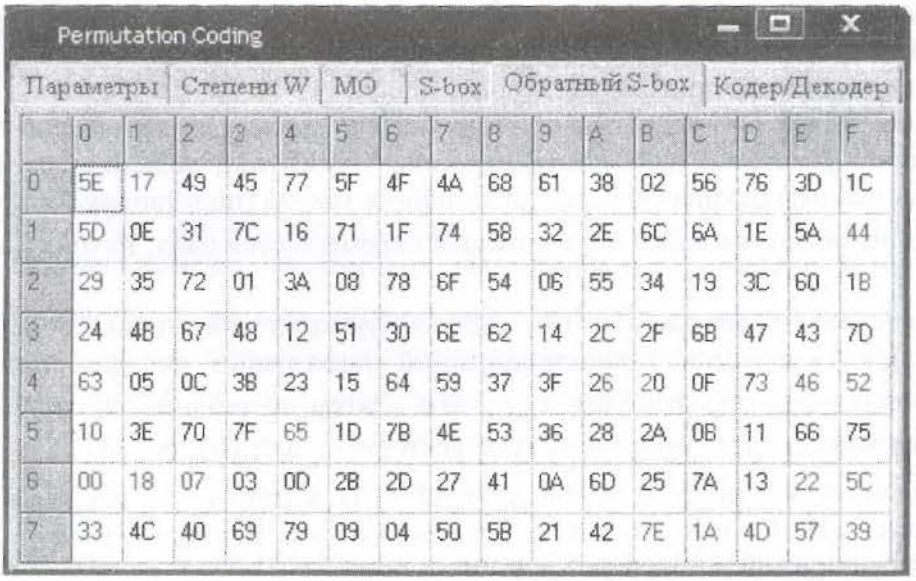
Панель результатов функции S-бокса выводится на экран монитора после нажатия на кнопку <S-box> базового интерфейса (рис. 2). На рис. 5 показан вариант S-бокса, параметры которого указаны на панели базового интерфейса.

Обратный S-бокс строится на основании инверсии таблицы прямого S-бокса. Идею инверсии поясним на конкретном примере. Выберем из таблицы S-бокса (рис. 5) байт 6F, расположенный на пересечении пятой строки и столбца C. В таблице обратного S-бокса элемент 5C надлежит разместить на пересечении шестой строки и столбца F. Аналогичным образом вычисляются остальные элементы таблицы. Таблица обратного S-бокса представлена на рис. 6.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	60	23	0B	63	76	41	29	62	25	75	69	5C	42	64	11	4C
1	50	5D	34	6D	39	45	14	01	61	2C	7C	2F	0F	55	1D	16
2	4B	79	6E	44	30	6B	4A	67	5A	20	9B	65	3A	66	1A	3B
3	36	12	19	70	2B	21	59	48	0A	7F	24	43	2D	0E	51	49
4	72	68	7A	3E	1F	03	4E	3D	33	02	07	31	71	7D	57	06
5	77	35	4F	58	28	2A	0C	7E	18	47	1E	78	6F	10	00	05
6	2E	09	38	40	46	54	5E	32	08	73	1C	3C	1B	6A	37	27
7	52	15	22	4D	17	5F	0D	04	26	74	6C	56	13	3F	7B	53

Рис. 5. Пример S-бокса



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	5E	17	49	45	77	5F	4F	4A	68	61	38	02	56	76	3D	1C
1	5D	0E	31	7C	16	71	1F	74	58	32	2E	6C	6A	1E	5A	44
2	29	35	72	01	3A	08	78	6F	54	06	55	34	19	3C	60	1B
3	24	4B	67	48	12	51	30	6E	62	14	2C	2F	6B	47	43	7D
4	63	05	0C	3B	23	15	64	59	37	3F	26	20	0F	73	46	52
5	10	3E	70	7F	65	1D	7B	4E	53	36	28	2A	0B	11	66	75
6	00	18	07	03	0D	2B	2D	27	41	0A	6D	25	7A	13	22	5C
7	33	4C	40	69	79	09	04	50	5B	21	42	7E	1A	4D	57	39

Рис.6. Пример обратного S-бокса

Функции кодирования/декодирования. С помощью функци кодирования/декодирования осуществляется одна из простейших форм зашифрования расшифрования за счет перестановки элементов текста, сохраняющегося в файл компьютера с любым расширением. Обращение к данной функции (рис. 7) происходи нажатием на индикатор <Кодер/Декодер>, находящийся в линейке функций базово интерфейса программного комплекса.

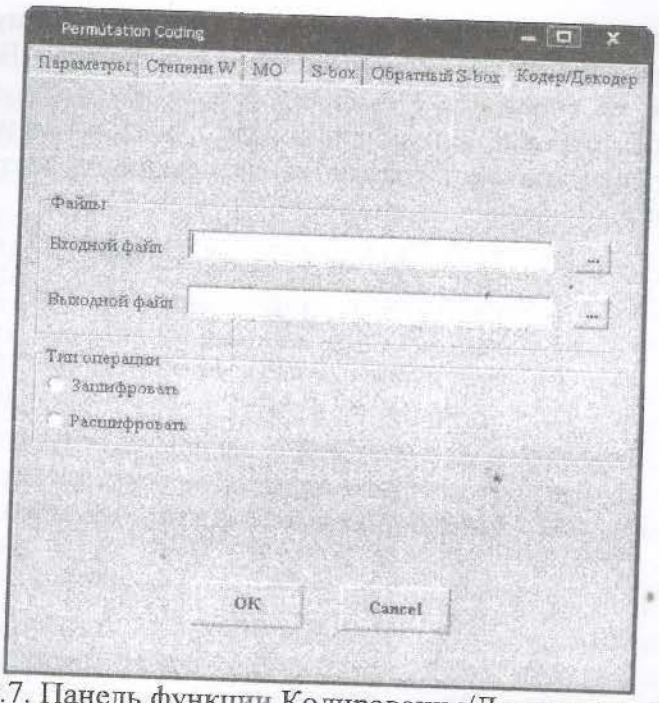


Рис.7. Панель функции Кодирования/Декодирования

Зашифруем в порядке эксперимента русскоязычный текст с расширением *doc объемом 209 Кбайт, гистограмма распределения частот которого приведена на рис. 8.

0	0	0	0	0	0	0	0
0	0	3890	0	0	3890	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
42720	35	284	0	0	0	0	2
4	4	4	0	3513	3336	2538	2
8	10	5	7	4	2	5	3
4	12	184	19	0	0	0	148
0	2	1	4	0	0	1	1
2	0	0	0	3	2	0	1
3	0	1	0	3	0	0	1
0	0	0	0	0	0	0	0
0	7	3	2	4	10	1	4
8	12	0	0	9	5	4	18
6	1	7	4	10	8	1	3
1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
326	63	290	35	118	93	18	71
203	0	117	36	173	288	355	202
41	601	213	52	53	41	9	39
0	0	0	0	3	85	42	157
11931	2344	6368	3041	4658	12105	1484	2533
10340	1424	4787	7369	4608	9631	16730	3877
5682	8008	10169	4524	73	1411	604	2573
1354	592	28	2767	3416	403	1044	3307

Рис.8. Гистограмма исходного текста

Энтропия H исходного текста составляет величину, равную 4,64162. Процесс зашифрования сводится к перестановке символов в пределах блоков. Если размер последнего блока открытого текста не совпадает с выбранной длиной блока, то он дополняется до полного пробелами. Эти пробелы при расшифровании становятся невидимыми на экране монитора. На рис. 9 приведена гистограмма зашифрованного теста с выше принятыми параметрами шифрования.

753	1346	720	815	495	821	457	445
553	745	947	832	455	621	912	579
635	1260	563	755	630	977	572	468
373	701	762	716	909	896	1513	979
920	1189	495	976	637	985	405	720
720	1253	735	1078	540	1166	637	865
898	1687	673	1150	1507	1721	977	1098
627	1250	716	1181	1181	1749	1664	1523
859	912	744	674	558	668	382	409
674	589	1432	1034	390	374	735	559
327	632	330	468	370	510	334	326
380	521	1077	822	495	545	954	669
481	884	356	623	464	677	289	478
778	1702	1070	1538	545	1508	640	986
737	1074	564	798	1014	1063	650	657
672	1296	972	1318	1043	1401	1166	1146
1268	1375	455	465	655	733	316	267
859	725	727	459	454	465	518	356
724	1054	357	379	484	594	290	250
432	572	510	436	684	607	845	564
1347	1203	593	592	882	912	440	452
1097	893	961	783	789	765	966	771
1054	1256	637	738	939	906	642	563
821	824	987	865	1254	1085	2141	1303
1358	1162	661	507	642	616	274	279
802	659	2866	2185	346	337	1173	955
845	839	429	394	614	566	311	262
625	608	2099	1440	513	473	1206	874
1155	983	699	518	679	615	354	364
887	1001	2302	1803	588	650	1256	947
1388	1212	1055	829	972	807	807	595
980	871	2073	1569	1084	816	1912	1210

Рис.9. Гистограмма зашифрованного текста

Как следует из шифрограммы, за один этап преобразования исходного текста криптопримитивом перестановки достигнута достаточно высокая степень «отбеливания» данных. Энтропия шифрограммы составляет 7.84, при максимуме, равно восьми.

Проведем более широкие экспериментальные исследования, сведенные в табл. 2, статистических свойств криптографического примитива для полного набора параметров, приведенных в табл. 1.

Таблица 2. Энтропия шифрограмм

Степень полинома (n)	$N = 64$	$N = 128$	$N = 256$	$N = 512$	Неприводимый полином (φ)
8	–	–	7.87748	7.40662	100111001
7	–	7.77179	7.34191	6.24669	10100111
6	7.71207	7.21223	6.24166	4.64162	1101101
5	6.90892	6.23566	4.64197	4.64162	101001
4	6.23083	4.64214	4.64197	4.64162	11001

Оценки энтропии получены для матриц преобразования $\hat{A}$, отвечающих СКГ 13, и аддитивных компонент $\alpha = 10101\ldots$ и $\beta = 01010\ldots$. В качестве входного использовался русскоязычный текстовый файл объемом 209 Кбайт.

На основании анализа данных табл. 2 приходим к таким заключениям. Во-первых, при фиксированной степени n неприводимого полинома φ_n, определяющего порядок перестановочной матрицы $\hat{A}$, и, соответственно, – число элементов перестановки, по мере возрастания длины блока N уменьшается энтропия шифрограммы H. И это естественно, поскольку с ростом N при фиксированном n увеличивается размер l переставляемых элементов блока. Когда l становится равным байту (а также – двум или четырем байтам для элементов перестановки типа слово или двойное слово), энтропия шифрограммы становится равной энтропии исходного текста с расширением *doc, поскольку сохраняются одинаковыми число элементов входного и выходного алфавитов. Зашифрование при этом сводится к перестановке отдельных символов, а также пар или четверок символов (элементов алфавита). Частота символов входного и выходного текстов при этом не меняются, что обеспечивает минимальное и постоянное значение энтропии H. По тем же самым причинам, во-вторых, для фиксированной длины блока N по мере роста степени НП n уменьшается энтропия H шифрограммы, достигая минимального значения, равного энтропии открытого текста, когда размер l окажется равным байту (а также двум или четырем байтам).

Заметим, что энтропия шифрограммы для вариантов, когда элементом замены выступает байтовская конструкция, будет несколько отличатся от энтропии исходного текста, так как недостающие символы блока в конце расширенного файла заполняются пробелами.

Выводы

Дружественный интерфейс разработанного программного комплекса предоставляет возможность гибкого изменения параметрами перестановки элементов шифруемых блоков, удобен в эксплуатации и может послужить основой построения других криптографических примитивов и шифров.

Список литературы

1.*Иванов М.А.* Криптографические методы защиты информации в компьютерных системах и сетях. – М.: КУДИЦ-ОБРАЗ, 2001. – 368 с.

2.*Шеннон К.Е.* Работы по теории информации и кибернетики.– М.: ИЛ, 1963. – 829 с.

3.*Шнайер Брюс.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2003. – 816 с.

4.*Мао В.* Современная криптография. Теория и практика. – М.: «Вильямс», 2005. – 768 с.

5.*Белецкий А.Я.* Преобразования Грея. / Белецкий А.Я., Белецкий А.А., Белецкий Е.А. Т. 1. Основы теории. – К.: Кн. Изд-во НАУ, 2007. – 412 с.

Разработан на алгоритмическом языке С++ программный продукт, обеспечивающий возможность комплексной оценки эффективности криптографических примитивов, осуществляющих перестановку элементов шифруемого блока, причем размер элемента перестановки и длина блока являются вариативными параметрами

Ключевые слова: программный комплекс, криптографический примитив, перестановка элементов блока

Розроблений на алгоритмічної мові С++ програмний продукт, що забезпечує можливість комплексної оцінки ефективності криптографічних примітивів, які забезпечують перестановку елементів блоку, що шифрується. При цьому розмір елемента перестанови та довжина блоку являються варіативними параметрами

Ключові слова: програмний комплекс, криптографічний примітив, перестановка елементів блоку

Developed at the algorithmic language C + + software product that provides the possibility of a comprehensive assessment of the effectiveness of cryptographic primitives implementing permutation of the elements of the encrypted block, and the size of the element permutation and block length are the variational parameters.

Keywords: software system, a cryptographic primitive, scrambling unit.

Поступила 10.09.2009