[2]. Korchenko A.A. The system development of fuzzy standards of network parameters, Zahist ìnformacìï, T.15, №3, 2013, pp. 240-246.
[3]. Korchenko A.A. The system of heuristic rules formation for network activity assessment, Zahist ìnformacìï, T.15, №4, 2013, pp. 353-359.
[4]. Stasiuk A.I., Korchenko A.A. A method of abnormality detection caused by cyber attacks in computer networks, Zahist ìnformacìï, 2012, №4 (57), pp. 129-134.
[5]. Korchenko A.G. The development of information protection systems based on the fuzzy sets, The theory and practical solutions, Kuev, 2006, 320 p.
[6]. Stasiuk A.I., Korchenko A.A. The basic model of parameters in attack detection (Identification) systems construction, Zahist ìnformacìï, 2012, №2 (55), pp. 47-51.

МЕТОД ФОРМУВАННЯ ЛІНГВІСТИЧНИХ ЕТАЛОНІВ ДЛЯ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Одним із рішень забезпечення інформаційної безпеки є системи виявлення вторгнень, засновані на аномальному принципі. Для побудови такого роду систем використовується метод виявлення аномалій, породжених кібератаками в інформаційних системах. У цьому методі процес формування різних еталонів досить трудомісткий і практично не формалізований, що знижує ефективність його використання. З метою компенсації цього недоліку пропонується метод, який базується на математичних моделях і методах нечіткої логіки та реалізується за допомогою шести базових етапів: формування підмножин ідентифікаторів лінгвістичних оцінок, формування базової матриці частот, формування похідної матриці частот, формування нечітких термів, формування еталонних нечітких чисел, візуалізація лінгвістичних еталонів. Метод дозволяє удосконалити процес формалізації отримання лінгвістичних еталонів параметрів для підвищення ефективності побудови відповідних систем виявлення вторгнень.

Ключові слова: кібератаки, аномалії, нечіткі еталони, метод формування лінгвістичних еталонів, системи виявлення вторгнень, системи виявлення аномалій, системи виявлення атак, виявлення аномалій в комп'ютерних мережах.

THE FORMATION METHOD OF LINGUISTIC STANDARDS CREATED FOR THE INTRUSION DETECTION SYSTEMS

One of the information security solutions are the intrusion detection systems based on the principle of anomaly. To develop such a kind of systems the anomaly detection method generated by cyberattacks in information systems is used. In this method, the process of formation of various standards is quite complicated and practically is not formalized, that reduces the efficiency of its use. In order to compensate for this drawback it is proposed the method which is based on mathematical models and methods of fuzzy logic and is implemented through the six basic stages: formation of subsets of linguistic assessments identifiers, formation of the basic matrix of frequencies, formation of a derivative matrix of frequencies, the formation of fuzzy terms, formation of fuzzy numbers, the visualization of linguistic standards. The method enables to improve the process of formalization of linguistic standards to increase the efficiency of the corresponding detection intrusion systems.

Keywords: cyber attacks, anomalies, fuzzy standards, the formation method of linguistic standards, intrusion detection systems, anomaly detection systems, attack detection systems, anomaly detection in computer networks.

Корченко Анна Олександрівна, кандидат технічних наук, доцент кафедри безпеки інформаційних технологій Національного авіаційного університету.
E-mail: annakor@ukr.net
Корченко Анна Александровна, кандидат технических наук, доцент кафедры безопасности информационных технологий Национального авиационного университета.
Korchenko Anna, PhD in Eng., Associate Professor of Academic Department of IT-Security, National Aviation University (Kyiv, Ukraine).

УДК 511.512

ПРОГРАММНО-МОДЕЛИРУЮЩИЙ КОМПЛЕКС КРИПТОГРАФИЧЕСКИХ AES-ПОДОБНЫХ ПРИМИТИВОВ НЕЛИНЕЙНОЙ ПОДСТАНОВКИ

Александр Белецкий, Анатолий Белецкий, Денис Навроцкий, Александр Семенюк

Любой итеративный блочный шифр должен содержать хотя бы один нелинейный примитив. Отсутствие нелинейных преобразований существенно снижает криптостойкость шифра, так как, сколько бы не было в раундах различных линейных примитивов, их совокупность может быть сведена к одному эквивалентному, что, как следствие, приводит к угрозе достаточно легкого взлома шифра. В работе изложена методика синтеза примитивов нелинейной

подстановки, основу построения которых составляет S-блок алгоритма Rijndael. Проведен сравнительный анализ, оцениваемый энтропией шифрограмм, трех классов примитивов. В первом классе примитивов шифруемые данные представлены одномерными бинарными векторами (байтами), во втором – в виде квадратных матриц восьмого порядка, и в третьем – пространственными матрицами (бинарными кубиками четвертого порядка). Обсуждаются возможности оптимизации параметров S-блоков, при которой достигается как минимум коэффициента корреляции взаимосвязи между входными и выходными переменными примитивов, так и максимум энтропии рассеивания откликов блоков.

Ключевые слова: *криптографический примитив, нелинейная подстановка, программный комплекс.*

I. Введение и постановка задачи.

Современные методы защиты информации в компьютерных сетях представляют собой математические преобразования, в которых сообщения рассматриваются как числа или алгебраические элементы в некотором пространстве [1]. С позиций теории сигналов и процессов зашифрование исходного (коррелированного, избыточного, сжимаемого) текста состоит в его «отбеливании», т.е. обращении в некоррелированную практически несжимаемую последовательность символов (элементов) шифрограммы с распределением вероятностей элементов выходного алфавита максимально близкой к дискретному равномерному распределению.

Криптостойкие системы могут быть построены путем многократного (итеративного) применения относительно простых криптографических преобразований (примитивов), в качестве которых К. Шенон предложил [2] использовать нелинейные подстановки (Substitution, или S - блоки) для *перемешивания* и линейные перестановки (Permutation, или P-блоки) для *рассеивания* шифруемых данных. Схемы, реализующие эти преобразования, называют SP - сетями.

В алгоритмах шифрования S - блоки осуществляют табличную подстановку, в результате которой $n-$битная группа входных символов $x \in X$ преобразуется в $n-$битную группу выходных символов $y \in Y$. В компьютерных шифрах первого поколения, таких, например, как DES [3], или ГОСТ [4], S - блоки выполняли преобразования «полубайт в полубайт» (схема 4x4). Узлы нелинейной замены современных примитивов в шифрах второго поколения, к числу которых в первую очередь следует отнести AES шифр [5], делают на основе подстановки «байт в байт» (схема 8x8). S - блок AES/Rijndael алгоритма реализует преобразование

$$y = x_f^{-1} \cdot A + \beta, \qquad (1)$$

где $x^{-1}-$мультипликативно обратный (МО) элемент относительно умножения в поле $GF(2^8)$, порождаемом неприводимым полиномом (НП) восьмой степени $f = 100011011$; $A-$ циркулянтная матрица восьмого порядка

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}; \qquad (2)$$

и байт $\beta = 01100011-$аддитивная компонента.

Матричное умножение в примитиве (1) выполняется в кольце вычетов по $\bmod 2$, а сложение байтов – поразрядным сложением по $\bmod 2$.

Основная задача данной статьи состоит в разработке методики синтеза AES-подобных S - блоков, реализованных в виде пакета прикладных программ, обеспечивающих возможность оптимизации параметров примитивов по критерию максимума энтропии шифрограмм, порождаемых S-преобразованиями входных текстов. Определение энтропии приведено в п. IV.

II. Методика синтеза S – блоков.

Обобщим примитив нелинейной подстановки шифра AES (1), придав ему форму

$$y = (x + \alpha)_f^{-1} \cdot A + \beta, \qquad (3)$$

где $\alpha-$дополнительная аддитивная компонента преобразования. Примитив (3) является базовым для RSB симметричных блочных криптографических алгоритмов [6].

Согласно соотношению (1), основной операцией при составлении AES-подобных примитивов нелинейной подстановки является операция вычисления МО y значений переменных x по модулю выбранного НП f степени n

$$y = x_f^{-1}. \qquad (4)$$

Взаимосвязь переменных x и их МО x_f^{-1} в (4) принято отображать в виде табл. 1, на внешних осях координат которой откладываются ста-

ршая x_2 и младшая x_1 компоненты переменной $x = x_2 \| x_1$ (здесь $\|$ – знак конкатенации), а на их пересечении (затененные элементы табл. 1) расположены МО значения y переменной x.

Таблица 1

Форма отображения взаимосвязи МО величин

	$x_1^{(0)}$	$x_1^{(1)}$	…	$x_1^{(j)}$	…	$\rightarrow x_1$
$x_2^{(0)}$						
$x_2^{(1)}$						
⋮						
$x_2^{(i)}$				$(x_2^{(i)} \| x_1^{(j)})_f^{-1}$		
⋮						
$\downarrow x_2$						

Мультипликативные обратные значения y переменной x по модулю НП f могут быть вычислены с помощью расширенного алгоритма Евклида [7]. Ниже приводится более простой способ формирования таблицы МО, изложенный в [8].

Суть этого метода состоит в следующем. Пояснения будем иллюстрировать числовыми примерами, выбрав в качестве НП полином четвертой степени $f = 10011$. Множество вычетов по $\mathrm{mod}\, f$, как раз и составляющих полное множество переменных x, представляют собой четырехбитные векторы (коды) от 0000 до 1111, которые там, где это окажется более удобным, будем записывать в виде двухразрядных четверичных чисел.

Этап 1. На этом этапе вычисляется мультипликативная группа максимального порядка (МГМП), содержащая $2^n - 1 = 15$ ненулевых вычетов (так как $n = 4$), представляющая собой последовательность степеней (от 0 до 15) примитивного образующего элемента (ОЭ) группы, (обозначим его ω), по $\mathrm{mod}\, f$. Поскольку выбранный НП f является примитивным, то минимальный двоичный ОЭ мультипликативной группы $\omega = 10$. Запишем $(k+1)$–ю степень ОЭ ω в общем виде очевидным соотношением:

$$\omega^{k+1} = \omega^k \cdot \omega \,(\mathrm{mod}\, f). \qquad (5)$$

Следовательно, согласно (5), если $\omega = 10$, то ω^{k+1} образуется в результате сдвига ω^k на один разряд влево. Если при этом старшая единица вектора ω^{k+1} перемещается в n–й разряд (разряды нумеруются справа налево, начиная с номера 0), то этот вектор приводится к остатку по $\mathrm{mod}\, f$. В табл. 2 сведена МГМП для выбранных параметров f и n. Приведем табл. 2 к форме табл. 1, причем степени и данные представим в четверичной форме (табл. 3), заменяя x на степени $k = k_2 \| k_1$.

Таблица 2

Мультипликативная группа над $f = 10011$ и $\omega = 10$

k	3	2	1	0		k	3	2	1	0
0	0	0	0	1		8	0	1	0	1
1	0	0	1	0		9	1	0	1	0
2	0	1	0	0		10	0	1	1	1
3	1	0	0	0		11	1	1	1	0
4	0	0	1	1		12	1	1	1	1
5	0	1	1	0		13	1	1	0	1
6	1	1	0	0		14	1	0	0	1
7	1	0	1	1		15	0	0	0	1

Таблица 3

Степени ОЭ $\omega = 10$ по $\mathrm{mod}\, f = 10011$

	0	1	2	3	$\rightarrow k_1$
0	01	02	10	20	
1	03	12	30	23	
2	11	22	13	32	
3	33	31	21	01	
$\downarrow k_2$					

Этап 2 предполагает формирование таблицы МО величин, отвечающих соотношению (4). Обратим внимание на то, что числа в ячейках табл. 3, связанных двунаправленными стрелками, являются мультипликативными обратными. В самом деле, рассмотрим, например, пару $32 \leftrightarrow 03$. Два числа являются МО, если их произведение в кольце вычетов по выбранному модулю, в рассматриваемом примере таковым является НП $f = 10011$, равно 1. Выполнив элементарные вычисления, убеждаемся в том, что числа 32 и 03 действительно являются мультипликативными обратными, также как взаимно обратными являются все симметрично расположенные числа в табл. 3.

Доказательство взаимно обратной связанности чисел, симметрично расположенных в таблицах степеней примитивных ОЭ ω по модулю произвольных НП f (примером такой таблицы является табл. 3) можно провести в общем виде достаточно простым математическим приемом.

В самом деле, пусть x и y – двоичные числа такие, что $x = \omega^k (\mathrm{mod}\, f)$ и $y = \omega^l (\mathrm{mod}\, f)$.

Число y обратно x по модулю f, т.е. $x \cdot y = \omega^k \cdot \omega^l (\mathrm{mod}\, f) = 1$, если только

$$k + l = 2^n - 1, \qquad (6)$$

где + есть оператор арифметической суммы.

Введем двоичное изображение

$$(2^n - 1)_2 = [1]^n, \qquad (7)$$

правая часть которого представляет собой n подряд записанных единиц.

Разрешая формально равенство (6) относительно l и, принимая во внимание (7), имеем

$$l = [1]^n - k.$$

В двоичной модулярной арифметике операцию вычитания можно заменить сложением, что приводит окончательному выражению

$$l_2 = [1]^n \oplus k_2, \qquad (8)$$

в котором $\oplus$ есть оператор поразрядного сложения по модулю 2, а нижний индекс, как и в (7), означает основание системы счисления.

Проиллюстрируем порядок вычисления МО величин по формуле (8), обратившись к табл. 3. Выберем, для примера, число $x = 31_4$. Двоичная степень ОЭ $\omega = 10$, которая по модулю $f = 10011$ образует данное число, составляет $k = 31_4 = 1101_2$. Двоичная степень l МО числа x определяется соотношением (8), согласно которому

$$l_2 = 1111 \oplus 1101 = 0010 = 02_4.$$

Вычисленной по формуле (8) степени l_2 (или $l_4 = 02$) в табл. 3 соответствует число 10_4, являющееся, как это легко проверить, мультипликативно обратным числу 31_4. На этом примере убеждаемся также в том, что пара взаимно обратных чисел располагается в симметрично расположенных ячейках таблицы степеней ОЭ ω по модулю выбранного НП f.

Следуя изложенному в данном разделе правилу, легко заполнить таблицу МО величин (табл. 4). В S-блоках нуль переходит в нуль, также как 1 переходит в 1 (по определению).

Таблица 4

Мультипликативные обратные числа по $\bmod f = 10011$

	0	1	2	3	$\rightarrow x_1$
0	00	01	21	32	
1	31	23	13	12	
2	33	02	30	11	
3	22	10	03	20	
$\downarrow x_2$					

Рассмотрим, в качестве примера, числа 10 и 31, связанные в табл. 3 двунаправленной стрелкой, т.е. являющиеся взаимно обратными числами. Эти числа заносят в табл. 4 следующим образом. В элемент табл. 4, находящийся на пересечении строки $x_2 = 1$ и столбца $x_1 = 0$ записывается число 31, являющееся мультипликативным обратным числу $x = x_2 \| x_1 = 10$. Аналогичным образом, в элемент табл. 4, расположенный на пересечении координат $x_2 = 3$ и $x_1 = 1$ (для краткости координаты переменной $x = x_2 \| x_1$ будем обозначать (x_2, x_1)), вписывается число 10. Подобным образом заполняются все оставшиеся ячейки таблицы.

На основании данных табл. 4 отобразим график зависимости (4), показанный на рис. 1. График взаимосвязи числа x и его МО значения y наглядно иллюстрирует, что эта зависимость является нелинейной.

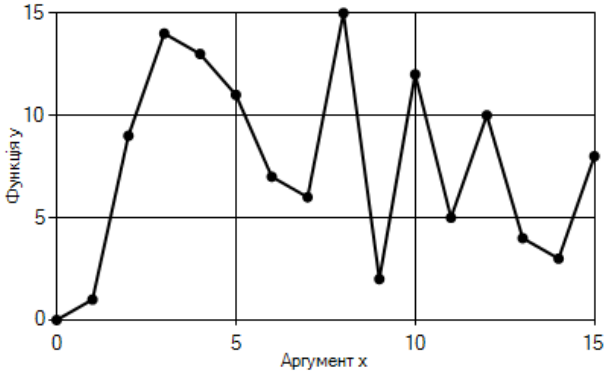

Рис. 1. Полигон распределения для МО (4)

Этап 3 достаточно простой и сводится к вычислению преобразования

$$y = (x + \alpha)_f^{-1}. \qquad (9)$$

Согласно (9) если $x = 0$, то y становится равным МО компоненты α. Пусть, для примера, $\alpha = 1011_2 = 23_4$. Четверичному числу 23 в табл. 4 отвечает МО значение 11, которое надлежит записать в элементе табл. 5 с координатами $(0, 0)$. Данная таблица предназначена для записи результатов вычислений преобразования (9). В клетке табл. 5 с координатами $(0, 1)$, стоящей справа от клетки с координатами $(0, 0)$, следует поместить МО алгебраической суммы (т.е. с поразрядным переносом) $x + \alpha = 01_4 + 23_4 = 30_4$. Числу $x_2 \| x_1 = 30_4$, согласно табл. 4, соответствует МО значение 22, размещаемое, как это уже отмечалось выше, в ячейке табл. 5 с координатами $(0, 1)$.

Из приведенных пояснений следует, что табл. 5 образуется в результате кругового сдвига элементов табл. 4, причем в качестве «лидера» цепочки сдвига следует выбрать МО аддитивной

компоненты α, равное 11, которое перемещается в начало табл. 5.

Таблица 5

Мультипликативные обратные суммы $x+\alpha$

	0	1	2	3	$\rightarrow x_1$
0	11	22	10	03	
1	20	00	01	21	
2	32	31	23	13	
3	12	33	02	30	
$\downarrow x_2$					

Этап 4 также является тривиальным и предполагает вычисление функции

$$y=(x+\alpha)_f^{-1}\cdot A. \qquad (10)$$

Матрица A должна быть невырожденной. Выберем в качестве такой матрицы, для примера, обобщенную примитивную $(0,1)-$*матрицу Галуа*, синтезированную по методу *диагонального заполнения.*

Суть алгоритма синтеза матриц Галуа, элементы которых $\alpha_{i,j}$, $i,j=\overline{1,n}$, принадлежат, в общем случае, расширенному полю $GF(2^n)$, заключается в следующем [9]. Пусть $\theta-$ образующий элемент матрицы, в качестве которого может быть выбран любой примитивный элемент поля $GF(2^n)$, порождаемого НП f_n. ОЭ θ записывается в нижней (первой) строке матрицы A. Элементы строки, расположенные левее θ, заполняются нулями. Последующие строки матрицы A (по направлению снизу вверх) образуются круговой прокруткой по часовой стрелке предыдущих строк матрицы. Если при этом левый элемент прокручиваемой строки равен 1, то выполняется обычный сдвиг строки на один разряд влево, а в правый освободившийся элемент строки записывается 0. При этом разрядность строки становится на единицу больше порядка матрицы. Векторы, отвечающие таким строкам, приводятся к остатку по модулю НП f и, тем самым, длина строки возвращается к порядку, равному порядку матрицы n. Выбрав образующий элемент $\theta=101$ и НП $f=10011$, получим

$$A=\begin{pmatrix}1&1&1&0\\0&1&1&1\\1&0&1&0\\0&1&0&1\end{pmatrix}. \qquad (11)$$

Перемножив элементы ячеек табл. 5 на матрицу (11), составим табл. 6.

Напомним, что двоичные матричные преобразования в (10) выполняются в кольце вычетов по модулю 2, т.е. все результаты вычислений приводятся к остатку по $\bmod 2$. Отметим, кроме того, что матрица A совсем необязательно должна быть примитивной. Достаточно того, чтобы она была невырожденной.

Таблица 6

Результаты вычислений по формуле (10)

	0	1	2	3	$\rightarrow x_1$
0	02	10	13	33	
1	32	00	11	23	
2	03	30	01	20	
3	31	12	22	21	
$\downarrow x_2$					

Указанное свойство гарантировано доставляется матрицам, синтезированным предложенным выше методом диагонального заполнения, причем в качестве образующих элементов θ матриц A могут быть выбраны совсем не обязательно примитивные элементы поля $GF(2^n)$, порождаемого выбранным неприводимым полиномом f.

Этап 5 завершает вычисления в соотношении (3) и, тем самым, завершает построение S-блока. На этом этапе все элементы табл. 6 поразрядно суммируются (без переносов) по $\bmod 4$ с аддитивной компонентой β, заданной в четверичной системе счисления.

Пусть, для примера, $\beta=0110_2=12_4$. Результаты поразрядного суммирования компоненты β с элементами табл. 6 представлены в табл. 7.

Таблица 7

Финальные результаты вычисления S-блока

	0	1	2	3	$\rightarrow x_1$
0	10	02	01	21	
1	20	12	03	31	
2	11	22	13	32	
3	23	00	30	33	
$\downarrow x_2$					

Таблицы $S-$блоков применяются на этапе зашифрования данных. Преобразования, осуществляемые $S-$блоком, представим общим соотношением

$$y=S(x), \qquad (12)$$

где $x=x_2\,||\,x_1=(x_2,x_1)$ – аргумент (или вход), а $y=y_2\,||\,y_1=(y_2,y_1)$ – функция (отклик) S-блока.

Способ определения отклика S-блока, т.е. переменной y в соотношениях (3) и (12), для

аргумента $x=(2,3)$ иллюстрируется в табл. 7 с помощью стрелок.

При расшифровании криптограммы восстанавливается переменная x по значению y, т.е. выполняется преобразование, обратное преобразованию (12), которое отобразим выражением

$$x = S^{-1}(y). \qquad (13)$$

На основании данных табл. 7 составим табл. 8 обратного преобразования, соответствующего соотношению (13). Перемещение выбранного в табл. 7 элемента $y=(3,2)$ в ячейку $x=(2,3)$ табл. 8 показано стрелками.

Таблица 8

Обратный S-блок

	0	1	2	3	$\rightarrow y_1$
0	31	02	01	12	
1	00	20	11	22	
2	10	03	21	30	
3	32	13	23	33	
$\downarrow y_2$					

III. Программный пакет «Синтез S-блока».

Все вычисления, приведенные в п. II статьи, включая построение графиков различных преобразований, реализованы в компьютерной программе, интерфейс которой показан на рис. 2.

Окна интерфейса (рис. 2) «загружены» теми же значениями параметров, которые выбраны для этапов синтеза S-блока, изложенных в п. II. Непосредственно под матрицей A расположен «светофор», индицирующий состояние матрицы: желтый цвет индикатора указывает на то, что матрица не определена; если загорается лампочка зеленого цвета, то это означает, что матрица невырожденная, а красного – вырожденная.

Кроме основного способа синтеза матриц A по методу диагонального заполнения, предусмотрена возможность (нажатием курсора) инверсии любого элемента матрицы, обеспечивая тем самым возможность «ручного ввода».

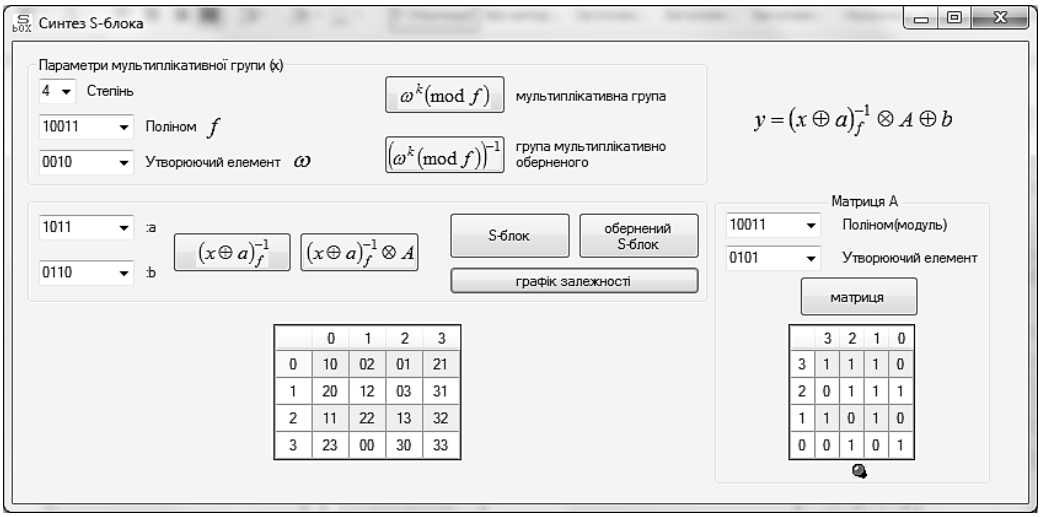

Рис. 2. Интерфейс программного пакета «Синтез S-блока

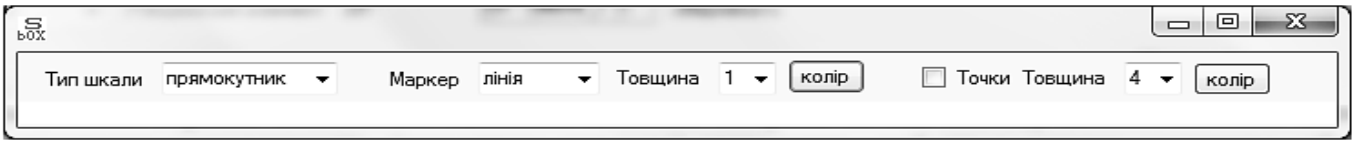

Рис. 3. Сервисы графического модуля

На рис. 3 показаны сервисы, предоставляемые пакетом посредством клавиши «Графік залежності». Сервис «Тип шкалы» дает возможность выводить графики в форме прямоугольника или квадрата. Сервис «Маркер» позволяет представить полигоны распределения выходных данных S-блоков в виде линий (пример показан на рис. 1), точек (рис. 4) или столбцов (рис. 5). Выбором сервиса «Точки» на границах линий устанавливаются дополнительные точки (рис. 1, 5), доставляющие лучшее восприятие графика. Кроме того, имеется возможность выбрать как толщину, так и цвет выводимых элементов графики.

IV. Программный пакет «Sub Universal».

Интерфейс пакета представлен на рис. 6. Программа выполняет зашифрование (расшифрование) произвольного входного (выходного) текста, адрес которого вносится в окошко «Вход-

ной (Выходной) файл». Шифрование текста осуществляется примитивом нелинейной замены байтов (S-блоком), который реализуется преобразованием (3). Параметризация S-блока производится выбором НП восьмой степени f и аддитивных компонент α и β, осуществляемых прокруткой в соответствующих окнах.

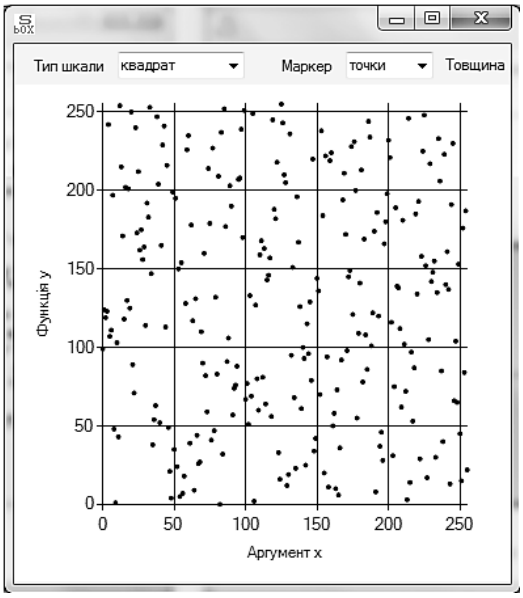

Рис. 4. Точечный полигон распределения S-блока (3) (параметры: $f_8 = 0\text{x}11\text{B}$, $\alpha = 0\text{x}0$, $\beta = 0\text{x}63$, $A - (2)$)

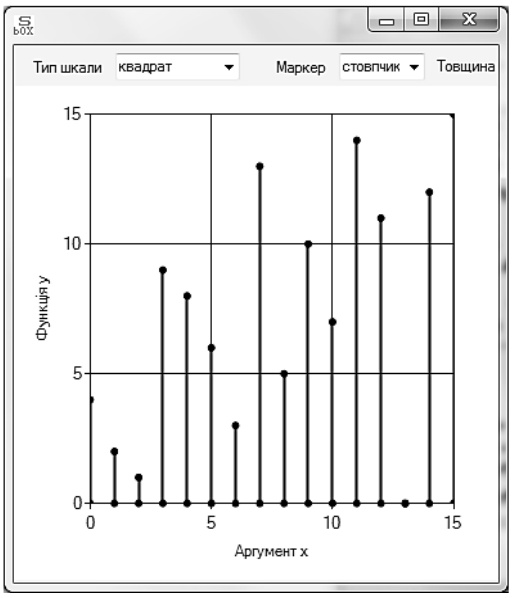

Рис. 5. Столбцовый полигон распределения S-блока (параметры блока определены в п. II)

Невырожденные матрицы A формируются автоматически нажатием на клавишу «Генератор матриц». Алгоритм синтеза матриц заключается в следующем. С помощью генератора псевдослучайных последовательностей элементы матрицы заполняются двоичными числами. Если при этом окажется, что матрица вырожденная (т.е. ее определитель по модулю 2 равен нулю), то выполняется очередное заполнение матрицы. Обновление элементов матрицы прекращается, как только ее определитель станет равным единице. Предусмотрен также ручной ввод матриц, который выполняется точно таким же способом, как и в программе «Синтез S-блока».

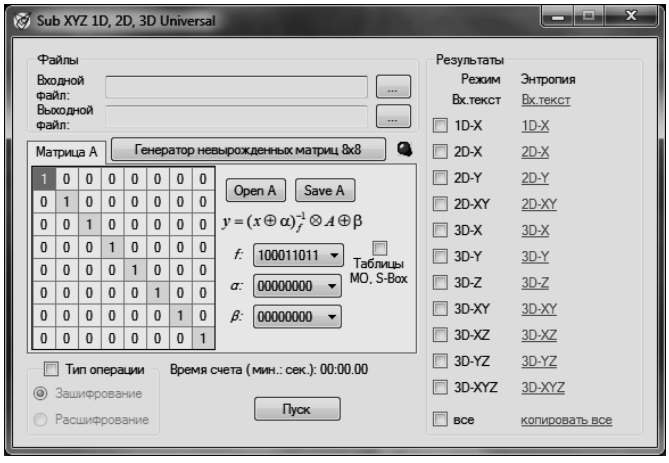

Рис. 6. Интерфейс программного пакета «Sub Universal»

Шифрование текстов выполняется в одном из трех пространств: 1D, 2D или 3D. В 1D пространстве шифруемые данные сохраняются в обычной форме одномерных векторов, обозначаемых X. Для этого пространства процесс S-преобразования данных (байтов) будем называть режимом 1D-X. В пространстве 2D исходный текст упаковывается в квадратные матрицы восьмого порядка таким образом, что в каждую строку матрицы заносится отдельный байт текста. Шифрование текста (S-преобразование байтов матриц данных) может осуществляться или последовательно по строкам матриц (режим 1D-X), или по ее столбцам (режим 1D-Y). И, наконец, в пространстве 3D как входные данные, так и шифрограммы представлены в форме кубиков третьего порядка 4x4x4 (рис. 7).

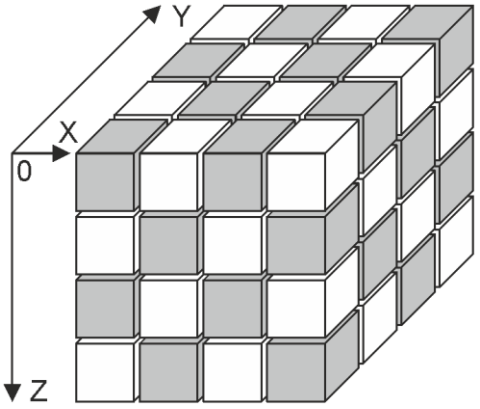

Рис. 7. Контейнер данных в пространстве 3D

Каждый элемент кубика содержит 1 бит информации. S-преобразованию подлежат байты, образуемые конкатенацией двух полубайтов, коллинеарных оси X (в режиме 3D-X), оси Y (в режиме 3D-Y) или Z (в режиме 3D-Z). Способ объединения полубайтов кубика в байты поясним, обратившись к рис. 8, в котором кубик данных представлен своими сечениями по оси X.

Элементы кубика (в плоскости с осями X, Y), примыкающие к оси X, верхние грани которых на рис. 8 закрашены черным цветом, образуют старший полубайт, а смежные с ними светлые элементы верхней грани кубика – младший полубайт первого S-преобразуемого байта в режиме 3D-X. Второй байт составляется аналогичным образом из оставшихся элементов верхней грани кубика данных. Для формирования третьего байта в режиме 3D-X берутся элементы кубика, расположенные ниже элементов первого байта и т.д. Естественно, что предлагаемая последовательность шагов, выполняемых при шифровании данных в режиме 3D-X, совпадает с последовательность операций, выполняемых для упаковки в кубики байтов исходного текста.

Таким образом, приведенный в предыдущем абзаце способ S-преобразования в режиме 3D-X можно сформулировать следующим образом. Первый шифруемый в режиме 3D-X текстовый байт составляется из полубайтов верхней грани кубика, расположенных на плоскости XY (рис. 7 или 8), коллинеарных оси X. Причем старший полубайт примыкает непосредственно к оси X. Второй байт, шифруемый в режиме 3D-X, также образуется из полубайтов, расположенных в плоскости XY верхней грани кубика и коллинеарных оси X, но смещенных по оси Y на два элемента относительно расположения полубайтов первого байта. Следующие пары байтов выбираются по схеме, подобной уже описанной, но со смещением вниз по оси Z на один элемент каждый раз, как только переходим к формированию очередной пары байтов.

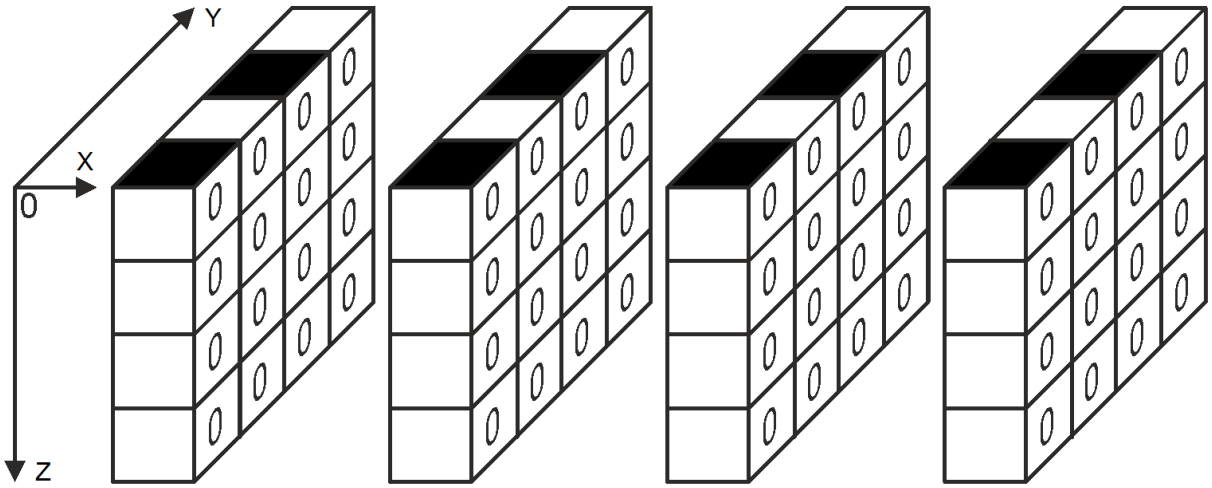

Рис. 8. Сечения кубика данных

Отобразим алгоритм выбора байтов в режиме 3D-X в виде такой символической схемы.

$$3D\text{-}X \in S(XY) \,||\, X \nearrow Y \downarrow Z, \qquad (14)$$

где обозначено: S – плоскость (в данном случае заданная осями координат X и Y); $||$ – символ коллинеарности; стрелки $\nearrow$ и $\downarrow$ указывают направление перемещения по осям Y и Z соответственно.

Используя символику (14), составим алгоритм выбора байтов данных в режимах 3D-Y и 3D-Z.

$$3D\text{-}Y \in S(YZ) \,||\, Y \downarrow Z \rightarrow X, \qquad (15)$$

$$3D\text{-}Z \in S(ZX) \,||\, Z \rightarrow X \nearrow Y. \qquad (16)$$

В соотношениях (14)-(16) легко просматривается циклический сдвиг на один символ влево как осей X, Y и Z, так и направлений перемещений $\nearrow$, $\downarrow$ и $\rightarrow$.

Кроме одномерных S-преобразований в кубиках (3D пространстве) можно выполнять также двумерные S-преобразования в режимах 3D-XY, 3D-YZ или 3D-ZX. Например, режим 3D-YZ означает, что над кубиком сначала выполняется преобразование (15), а затем преобразование (16). И, наконец, в режиме 3D-XYZ над кубиком последовательно проводятся все три преобразования по соотношениям (14), (15) и (16).

Для каждого из выбранных режимов S-преобразований в дискретных пространствах представления данных 1D-3D программой Sub Universal выполняется расчет энтропии шифрограммы для входного текста, адрес которого указан в окне «Входной файл».

Энтропия H оценивается в битах по формуле Шеннона

$$H = -\sum_{i=0}^{255} p_i \log_2 p_i, \quad p_i = n_i / N, \; N = \sum_{i=0}^{255} n_i,$$

где n_i – число байтов входных символов (или символов шифрограммы), состояние которых (байтов) определяется двоичным эквивалентом десятичного числа i.

На рис. 9 приведены результаты вычисления энтропии шифрограмм (размещенных в правой части интерфейса), полученных S-преобразованием русскоязычного текста объемом 208 Кбит.

V. Анализ эффективности и оптимизация параметров S-блоков.

Согласно компьютерным расчетам, показанных на рис. 9, энтропия шифрограмм S-преобразований в режимах 1D-X, 2D-X и 3D-X совпадает с энтропией входного текста. Этого следовало ожидать, поскольку последовательность байтов входного текста совпадает с последовательностью байтов, упакованных в матрицы восьмого порядка (в пространстве 2D), или кубики четвертого порядка (в пространстве 3D).

Как следует из приведенных расчетов, в случае одномерных S-преобразований, под которыми понимается преобразования байтов, коллинеарных одной из осей пространства представления данных, энтропия шифрограмм возрастает, если ось направления шифрования X поменять на Y или Z. Естественно, что энтропия шифрограмм каждого двумерного S-преобразования превышает энтропию шифрограмм любого из одномерных S-преобразований. Максимального значения энтропия достигает, когда S-преобразование осуществляется по всем трем осям кубиков данных в пространстве 3D.

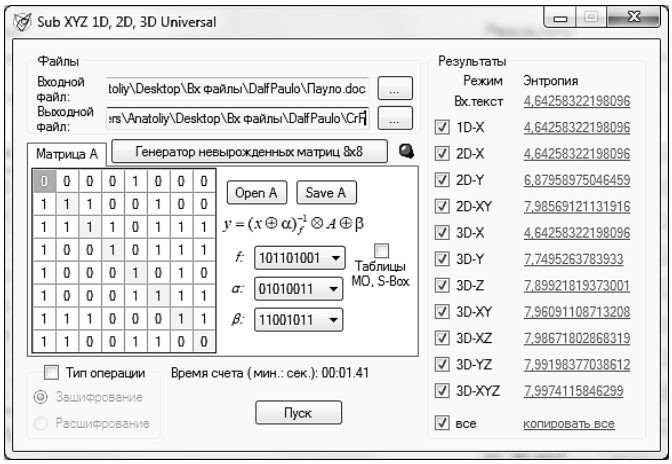

Рис. 9. Результаты расчета энтропии шифрограмм входного текста

В данном разделе работы попробуем ответить на вопрос о существовании (или отсутствии) оптимальных значений параметров, к числу которых будем относить параметры S-блока f, α, β и матрицу преобразования A, доставляющих максимум энтропии H шифрограмам входных текстов.

Важно подчеркнуть, что задачи, основанные на многопараметрической оптимизации, не имеют универсального способа решения [10]. Можно, например, воспользоваться методом оптимизации, который основан на полном переборе всех возможных значений параметров преобразования на интервалах их дискретного определения, которое характерно для рассматриваемой задачи. Аналогом данного метода служит «лобовая атака» на криптосистему, которая сводится к полному (тотальному) перебору всех состояний ключа. Совершенно очевидным является тот факт, что «лобовая атака» не всегда может быть осуществима практически. Проблема состоит в том, что оптимизация по данному методу зачастую упирается в недопустимо громадные вычислительные ресурсы, которые необходимы для реализации процесса определения оптимума.

В таком случае, как это зачастую практикуется, например, в криптоанализе, прибегают к комбинированию формальных математических и интуитивно-логических приемов оптимизации. Далее мы как раз и воспользуемся таким способом, который позволит прийти к инженерному решению относительно выбора приемлемых значений параметров S-блока, представленных соотношением (3).

В табл. 9 приведены, в качестве примера, результаты расчетов энтропии шифрограмм русскоязычного текста объемом 208 Кбайт для параметров S-блока, принятых в алгоритме AES.

Таблица 9

Энтропии шифрограмм текста, формируемых S-блоком алгоритма AES

Режим	Энтропия
2D-Y	6.9672
2D-XY	7.9630
3D-Y	7.7562
3D-Z	7.9218
3D-XY	7.9742
3D-XZ	7.9831
3D-YZ	7.9958
3D-XYZ	7.9975

Как уже было отмечено выше, энтропия H входного текста совпадает с энтропией шифрограмм, образуемых S-преобразованиями этого тек-

ста в режимах 1D-X, 2D-X и 3D-X. Для сравнения в табл. 10 приведены оценки энтропий шифрограмм того же самого текста, что и при вычислении энтропии шифрограмм, формируемых S-блоком шифра AES (табл. 9).

Таблица 10

Энтропии шифрограмм по множеству десяти НП восьмой степени

Режим	1.11B	2.11D	3.12B	4.12D	5.139	6.13F	7.14D	8.15F	9.163	10.165
2D-Y	6,8916	6,9696	6,9885	6,9694	6,9908	6,7916	6,8775	6,9963	6,9445	6,9080
2D-XY	7,9884	7,9881	7,9786	7,9809	7,9908	7,9659	7,9841	7,9906	7,9764	7,9844
3D-Y	7,7920	7,7871	7,7509	7,7567	7,7901	7,7054	7,7887	7,7211	7,8196	7,8099
3D-Z	7,8644	7,8179	7,8190	7,9410	7,8729	7,8255	7,8534	7,7242	7,8523	7,7392
3D-XY	7,9679	7,9665	7,9432	7,9591	7,9690	7,9595	7,9680	7,9743	7,9631	7,9650
3D-XZ	7,9793	7,9804	7,9847	7,9869	7,9483	7,9654	7,9515	7,9775	7,9824	7,9716
3D-YZ	7,9949	7,9957	7,9924	7,9945	7,9947	7,9960	7,9930	7,9890	7,9949	7,9961
3D-XYZ	7,9977	7,9973	7,9973	7,9975	7,9962	7,9977	7,9973	7,9979	7,9978	7,9972

В качестве параметров S-преобразований приняты те, которые указаны на рис. 9. В колонках строки «Режим» таблицы 16-ричными числами обозначены первые 10 значений НП восьмой степени, причем слева от разделительной точки поставлен порядковый номер полинома.

Из сопоставления данных табл. 9 и 10 видно, что замена НП практически не оказывает сколько-нибудь существенного влияния на энтропию шифрограмм, формируемых S-блоком (3). Более того, как показали результаты численных расчетов, проведенных с помощью программы «Sub Universal», подобным же образом сказывается на поведении энтропии шифрограмм вариации аддитивными компонентами α, β и матриц преобразования A. Это означает, в частности, что допускается достаточно свободный выбор параметров S-блоков (f, α, β) и A в широком диапазоне.

Выводы. На основании проведенных исследований можно сформулировать следующее заключение. Во-первых, вряд ли можно считать обоснованным утверждение некоторых ученых о том, что разработчики шифра AES пришли к параметрам S-блока (неприводимому полиному $f = 100011011$, матрице A, заданной выражением (2), и аддитивной компоненте $\beta = 01100011$) в результате тщательной и скрупулезной оптимизации. Подтверждением сказанному служит, возможно, такой факт: выбранный для шифра AES неприводимый полином есть первый полином в списке из 30-ти НП восьмой степени, что может служить косвенным подтверждением случайности его выбора. И, во-вторых, рассевающие свойства AES-подобных S-блоков (3), оцениваемые энтропией формируемых ими шифрограмм (или коэффициентом корреляции вход/выходных переменных блоков), не являются чувствительными к параметрам преобразования. Но для шифров специального назначения эти параметры α, β и A могут выступать в качестве долговременных ключей, как это принято частично, например, в российском симметричном блочном шифре ГОСТ.

ЛИТЕРАТУРА

[1]. Мао В. Современная криптография. Теория и практика. / В. Мао – М.: «Вильямс», 2005. – 768 с.

[2]. Шеннон К.Е. Теория связи в секретных системах/К.Е. Шеннон – М.: Изд-во ИЛ,1963. – 829 с.

[3]. Data Encryption Standard (DES) – FIPS 46-3 [Электронный ресурс]: http://csrc.nist.gov/ publications/fips/fips46-3/fips46-3.pdf

[4]. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – Изд-во стандартов, 1989. – 18 с.

[5]. Advanced Encryption Standard (AES) – FIPS 197 [Электронный ресурс]: http: //csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[6]. Белецкий А.Я. Преобразования Грея: Монография в 2-х томах. Т. 2. Прикладные аспекты. / А.Я. Белецкий, А.А. Белецкий, Е.А. Белецкий. – К.: Кн. изд-во НАУ, 2007. – 644 с.

[7]. Смарт Н. Криптография. / Н. Смарт. – М: Техносфера, 2005. – 528 с.

[8]. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях. / М.А. Иванов. – М.: КУДИЦ-ОБРАЗ, 2001. – 368 с.

[9]. Белецкий А.А. Криптографические приложения обобщенных матриц Галуа и Фибоначчи / А.А. Белецкий // Захист інформації, 2013. – С. 128-133.

[10]. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации / И.В. Сергиенко – К.: Изд-во «Наук. думка», 1982. – 327 с.

REFERENCES

[1]. Mao Wenbo. Modern Cryptography. Theory and Practice, M.: «Viljams», 2005, 768 p.

[2]. Shannon C.E. Communications theory of secrecy systems, M.: Pbl. IL, 1963, 829 p.

[3]. Data Encryption Standard (DES) – FIPS 46-3 http://csrc.nist.gov/publications/fips/fips46-3/fips346-3.pdf
[4]. GOST 28147-89. The information processing system. Cryptographic Security. Algorithm cryptographic transformation, Standards Press, 1989, 18 p.
[5]. Advanced Encryption Standard (AES) – FIPS 197 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[6]. Beletsky A.Ja., Beletsky A.A., Beletsky E.A. Transformation of Gray. Monogr. in 2 vols. V. 2. Application aspects, K.: Book House NAU, 2007, 644 p.
[7]. Smart N. Cryptography: An Introduction., M: Tehnosphera, 2005, 528 p.
[8]. Ivanov M.A. Cryptographic methods of information security in computer systems and fields, M.: KUDIC-OBRAZ, 2001, 368 P.
[9]. Beletsky A.Ja. Cryptographic applications generalized matrix Galois and Fibonacci, Security Information, 2013, pp. 128-133.
[10]. Sergienko I.V. Mathematical models and methods for solving discrete optimization, K.: Book House «Naukova Dumka», 1982, 327 p.

ПРОГРАМНО-МОДЕЛЮЮЧИЙ КОМПЛЕКС КРИПТОГРАФІЧНИХ AES-ПОДІБНИХ ПРИМІТИВІВ НЕЛІНІЙНОЇ ПІДСТАНОВКИ

Будь якій ітеративний блочний шифр повинен мати хоча б один нелінійний примітив. Відсутність нелінійних перетворень істотно знижує криптостійкість шифру, тому як скільки б не було в раундах різних лінійних примітивів, їх сукупність може бути зведена до одного еквівалентного, що, як наслідок, призводить до загрози достатньо легкого зламу шифру. У роботі викладена методика синтезу примітивів нелінійної підстановки, основу побудови яких становить S-блок алгоритму Rijndael. Проведено порівняльний аналіз (за критерієм максимуму ентропії шифрограм) трьох класів примітивів. У першому класі примітивів дані, що шифруються, представлені одновимірними бінарними векторами (байтами), у другому - у вигляді квадратних матриць восьмого ступеня, і в третьому - просторовими матрицями (бінарними кубиками четвертого порядку). Обговорюються можливості оптимізації параметрів S-блоків, при яких досягається як мінімум коефіцієнта кореляції взаємозв'язку між вхідними та вихідними змінними примітивів, так і максимум ентропії розсіювання відгуків блоку.
Ключові слова: криптографічний примітив, нелінійна підстановка, програмний комплекс.

SOFTWARE-MODELING COMPLEX CRYPTOGRAPHIC AES-LIKE PRIMITIVES NONLINEAR SUBSTITUTIONS

Any iterative block cipher should contain one non-linear primitive at least. Lack of a nonlinear transformations significantly reduces the cryptographic cipher strength because of any combination of linear primitives could be reduced to the equivalent one, so as a consequence leads to the cipher compromise. In this paper method of nonlinear substitution primitives synthesis is proposed, grounded on S-box construction accordingly to Rijndael algorithm. Three different classes of primitives have evaluated with regard to cryptograms entropy. First class consists of primitives for which an encrypted data is represented by one-dimensional binary vector(bytes), for the second one – data represented by square matrices with an order of eight, for the third one- by three-dimensional matrices (binary fourth-order cubes). Provided possibilities of optimizing the S-box parameters in order to achieve both a minimum correlation coefficient between input and output variables of a primitives and the maximum block response dissipation entropy.
Keywords: cryptographic primitive, non-linear substitution, software package.

Белецкий Александр Анатольевич, младший научный сотрудник Национального авиационного университета.
E-mail: alexander.beletsky@gmail.com
Білецький Олександр Анатолійович, молодший науковий співробітник Національного авіаційного університету.
Beletsky Alexander, Junior Researcher of National Aviation University.

Белецкий Анатолий Яковлевич, доктор технических наук, профессор кафедры электроники Национального авиационного университета.
E-mail: abelnau@ukr.net
Білецький Анатолій Якович, доктор технічних наук, професор кафедри електроніки Національного авіаційного університету.
Beletsky Anatoly, doctor of Science, Professor of Department Electronics of National Aviation University.

Навроцкий Денис Александрович, аспирант кафедры электроники Национального авиационного университета.
E-mail: sg6336@yandex.ua
Навроцький Денис Олександрович, аспірант кафедри електроніки Нац. авіаційного університету.
Navrotskyi Denys, postgraduate student of Department Electronics of National Aviation University.

Семенюк Александр Иванович, студент кафедры электроники Национального авиационного университета.
E-mail: sovist9@mail.ru
Семенюк Олександр Іванович, студент кафедри електроніки Національного авіаційного університету.
Semenjuk Alexander, student of Department Electronics of National Aviation University.