АЛГОРИТМ ШИФРОВАНИЯ С ПЕРЕМЕННОЙ ФРАГМЕНТАЦИЕЙ БЛОКА

Олег Николаевич Жданов

Канд. ф.-м. наук, Сибирский государственный аэрокосмический университет, Красноярск, Россия

Артем Викторович Соколов

Канд. тех. наук, Одесский национальный политехнический университет, Одесса, Украина

АННОТАЦИЯ

В работе предложен алгоритм шифрования с динамическим изменением размеров криптографических примитивов в различных раундах шифрования. Показано, что разработанный алгоритм обладает хорошими стохастическими и криптографическими свойствами.

ABSTRACT

This paper proposes an encryption algorithm with dynamic resizing of cryptographic primitives in various rounds of encryption. It is shown that developed algorithm has good cryptographic and stochastic properties.

Ключевые слова: алгоритм шифрования, S-блок, фрагментация.

Keywords: encryption algorithm, S-box, fragmentation.

Как известно, многие современные алгоритмы шифрования используют замены по таблице и операцию сложения с раундовым ключом [9]. При этом важным является как размер заменяемого блока, так и качество таблицы замен, влияющих на показатели диффузии и конфузии.

Блок является объектом, над которым выполняется преобразование. Однако блок текста обладает своей структурой, в большей или меньшей степени наследует особенности текстов на естественном языке. Поэтому работа

с блоками неизменного, одного и того же размера во всех раундах может не гарантировать рассеяния таковых особенностей по шифртексту, и для повышения качества итогового криптопреобразования приходится либо усложнять раундовые операции с блоками либо увеличивать количество раундов.

В настоящей работе представлен алгоритм шифрования с динамическим изменением размеров криптографических примитивов в различных раундах.

Иными словами, мы предлагаем проводить зашифрование текста, применяя замены по таблицам разных размеров в различных раундах.

Генерация высококачественных S-блоков часто выполняется в поле Галуа $GF(2^k)$, $k \in \mathbb{N}$, где k — размер входного слова S-блока.

Для наиболее эффективной организации алгоритма шифрования с динамическим изменением размеров криптографических примитивов выбор длины входного блока алгоритма должен быть составным числом. Например, рассмотрим блок открытого текста длины $L = 120 = 2^3 \cdot 3 \cdot 5$ бит. Разбиение данного блока открытого текста в алгоритме шифрования может быть произведено различными способами, например, на сегменты удобной с вычислительной точки зрения длины $\sigma = 6,8,10,12,15,20,30$ бит, при этом в пределах одного раунда размер сегмента не меняется.

Для каждого сегмента блока длины p производится подстановка в соответствии с правилом, определенным S-блоком. Выбор S-блоков для каждого раунда происходит в соответствии с критериями криптографического качества [10], их синтез может быть произведен в соответствии с ранее разработанными методами [2,4,6,7,8,11,12]. После выполнения операции подстановки производится конкатенация $120/p$ полученных сегментов, мы снова имеем блок 120 бит.

Следующее преобразование—циклический сдвиг вправо на 13 бит. Далее производится еще одна сегментация полученных 120 бит на сегменты длины q бит, раундовый ключ также фрагментируется на сегменты длины q бит,

выполняется сложение сегментов текста и сегментов ключа по модулю 2^q. Результирующие сегменты снова объединяются в блок длины 120 бит, который и является результатом первого раунда шифрования. Перед началом второго раунда ключ (длины 120 бит) сдвигается вправо циклически на 11 бит.

Второй раунд выполняется по такой же схеме с (вообще говоря) другими значениями p и q.

Представим раунд зашифрования в общем виде с помощью рис. 1.

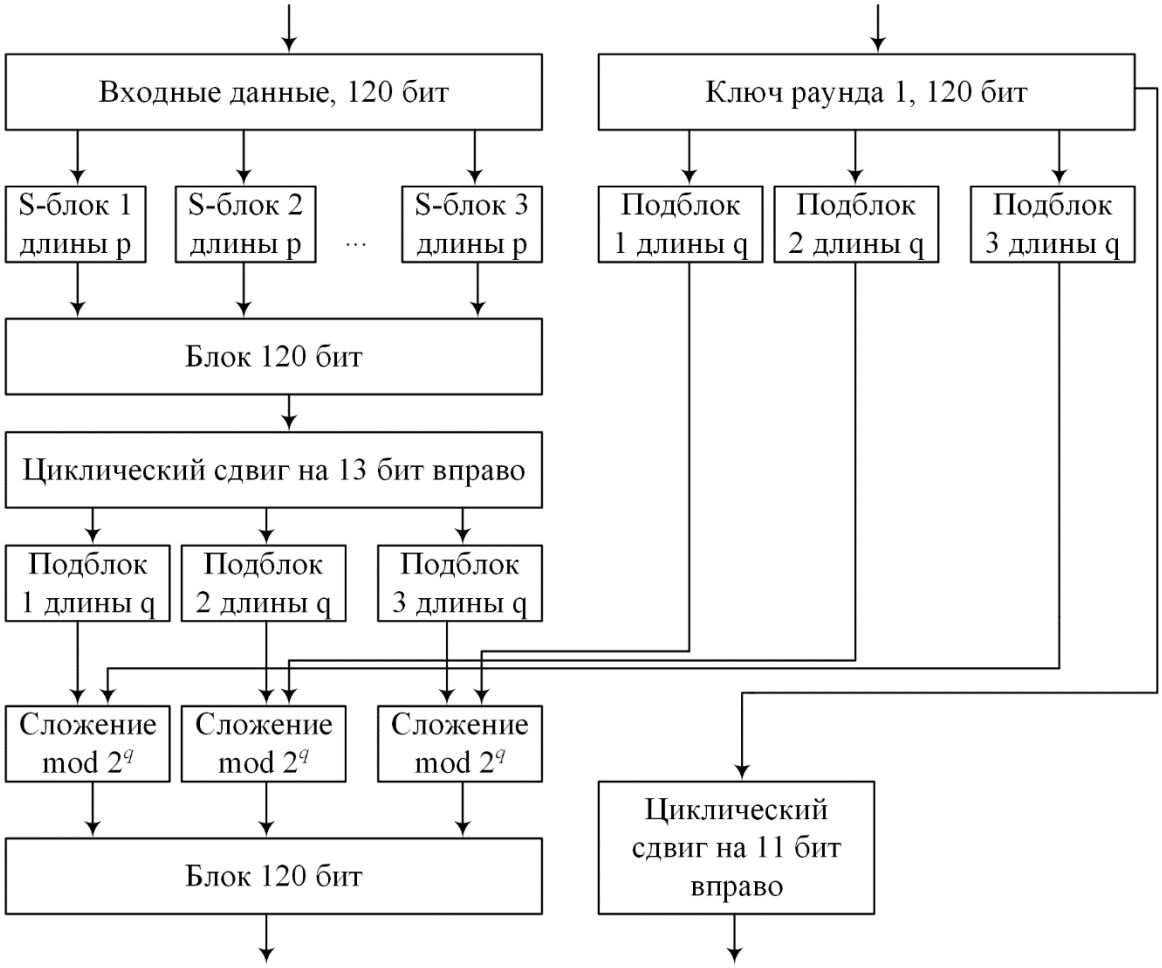

Рисунок 1. Раунд зашифрования

О таблицах замен. Возможны два подхода к их формированию. Первый: генерация оптимальных таблиц, например, по методике, изложенной в [7]. Полученные применением методики таблицы обладают идеальными матрицами коэффициентов корреляции $R = \|0\|$, а также соответствуют строгому

лавинному критерию. Заметим, что эти таблицы не обладают высокими значениями нелинейности.

Второй подход состоит в выборе многочлена, неприводимого над соответствующим полем Галуа [2,6]. Сегмент блока длиной p бит мы трактуем как набор коэффициентов многочлена над полем Галуа $GF(2^p)$ и тем самым ставим в соответствие данному сегменту многочлен, после чего находим обратный мулитипликативно обратный к данному многочлену элемент по модулю заданного неприводимого многочлена $f_i(x)$. Результат представляется в виде двоичного эквивалента, то есть выполняется преобразование, аналогичное первому этапу операции SubBytes алгоритма Rijndael. Однако при таких заменах мы не получаем таблицы с нулевой корреляционной матрицей.

Мы предлагаем совмещение двух указанных подходов: в нескольких раундах использовать оптимальные S-блоки подстановки, в последующих раундах используются S-блоки на основе мультипликативно обратных элементов.

Раунд расшифрования содержит подобные операции, лишь изменяется их порядок: сначала выполняется вычитание ключа из шифротекста под $\mathrm{mod}\, 2^q$ с параметром сегментации q, а потом подстановка с параметром сегментации p. Все циклические сдвиги вправо при расшифровании заменяются на циклические сдвиги влево. Раунд расшифрования схематически изображен на рис. 2.

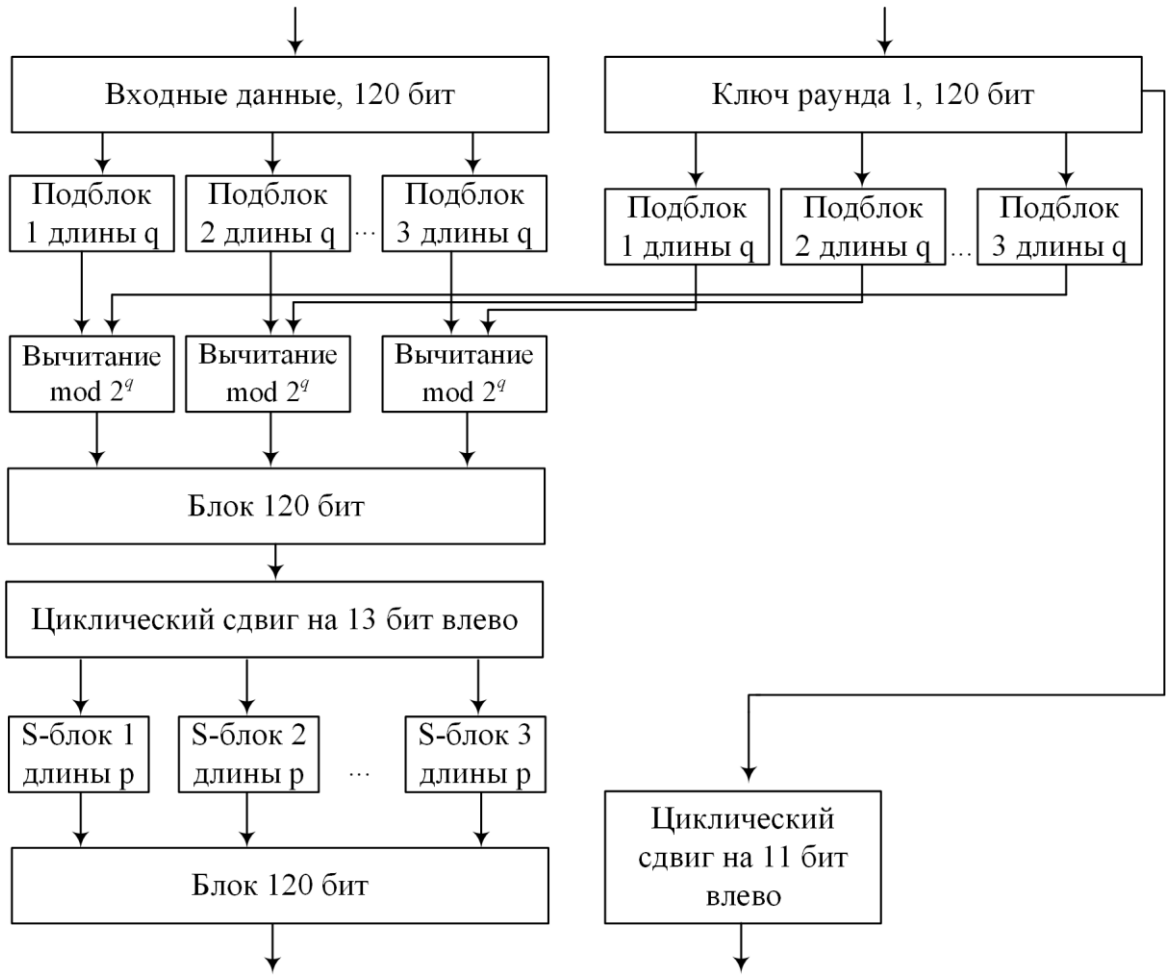

Рисунок 2. Раунд расшифрования

Конечно, все раундовые значения $\{p,q\}_i$, $i=1,2,..,$ применяются в обратном порядке следования для расшифрования.

Оценим число уровней защиты разработанного алгоритма. Очевидно, сам ключ может быть выбран 2^{120} различными способами. В случае, если выбраны параметры сегментации $\sigma = 6,8,10,12,15,20,30$ и 7 раундов шифрования, то выбор параметров $\{p,q\}$ можно произвести $(7\cdot 16)^7 = 221068140740608$ способами. Таким образом, число уровней защиты при несекретных таблицах замены составляет $\Psi = 221068140740608\cdot 2^{120} \approx 2.93\cdot 10^{50}$. Число уровней защиты может быть легко увеличено путем засекречивая используемых таблиц замен. В этом случае, учитывая объемы S-блоков [7] несложно добиться его значений многократно превышающих число Шеннона [3]. Кроме того, количество раундов легко может быть увеличено. Более того, ценой совсем

небольшого усложнения конструкции можно и количество раундов сделать псевдослучайным и неизвестным противнику.

Аспекты практической реализации. Вариант первый. У пользователей есть набор значений фрагментации и набор таблиц замен. Эти наборы фиксированы для каждой пары пользователей и секретны.

Вариант второй. Отправитель выбирает псевдослучайным образом параметры фрагментации и набор многочленов над соответствующими полями Галуа и передает эту информацию отправителю пользователю одновременно с зашифрованным текстом. Это можно сделать, например, используя схему вероятностного шифрования с применением BBS-генератора [1]. Поскольку в данном случае схема вероятностного шифрования будет использоваться не для передачи больших текстов, а лишь для передачи параметров, то минусы такой схемы не окажут существенного влияния на время и стоимость передачи сообщений.

Третий вариант. Отправитель и получатель выбирают параметры генератора псевдослучайной последовательности(например, BBS-генератора) и каждый получает такую последовательность, после чего на ее основе выбирают параметры фрагментации.

Приведем пример работы предложенного алгоритма шифрования.

*Раунд 1.*Разбиваем блок на сегменты по 6 бит, заменяем каждый сегмент по оптимально таблице [7]

$S_6 =$

S	0	1	2	3	4	5	6	7
0	34	30	04	62	02	71	41	56
1	66	05	53	52	27	35	61	17
2	51	43	15	65	06	63	37	36
3	72	07	33	24	54	50	60	00
4	70	74	22	44	31	42	16	01
5	45	26	12	13	75	67	57	21
6	03	11	25	55	23	46	76	77
7	47	32	64	73	10	14	40	20

. (1)

Выполняем конкатенацию, снова у нас 120 битовый блок. Сдвиг на 13 позиций вправо, складываем побитово с ключом. Ключ сдвигаем на 11 бит.

Раунд 2. Фрагментируем на сегменты по 10 бит, замена по оптимальной таблице, сгенерированной на основе (1) по алгоритму [7]. Выполняем конкатенацию. Сдвиг на 13 бит вправо. И фрагментируем на сегменты по 8 бит. Складываем с 8 битовыми сегментами ключа по модулю 2^8. Снова выполняется конкатенация.

Сдвиг ключа на 11 бит.

Для *следующих 5 раундов* выбраны наборы сегментации преобразования замены и сложения с ключом: $\{(p,q)\} = \{(8,15),(6,10),(4,20),(12,20),(8,30)\}$, выбраны многочлены

$$\begin{cases} f_6(x) = x^6 + x + 1; \\ f_8(x) = x^8 + x^4 + x^3 + x^2 + 1; \\ f_{10}(x) = x^{10} + x^3 + 1; \\ f_{12}(x) = x^{12} + x^6 + x^4 + x + 1, \end{cases} \qquad (2)$$

для соответствующих размеров входных данных. В нашем примере для простоты выбран один и тот же полином в третьем и седьмом раундах.

Пусть задан ключ

$$K = [11000001010000111100000111001111110100001010010110011100$$
$$11111010010110001000000001011001001011110101110001111011010011011].$$

Отметим, что для демонстрации хороших качеств нашего алгоритма мы специально выбрали заведомо слабый ключ. Покажем, что даже при таком ключе мы получим шифртекст, мало отличающийся от случайной последовательности.

Пусть задана псевдослучайная последовательность длины $N = 14400$, обладающая значительной избыточностью. Так, количество нулевых символов $K^0 = 11965$, и, соответственно, единичных $K^1 = 2435$, что соответствует многим информационным двоичным последовательностям, обладающим природной избыточностью, таким как: цифровые изображения, цифровое видео, текст и т.д. Начало последовательности имеет вид

$$P = [1000001000010000001010000000000001000010010010000000$$
$$0000010101000001010010000001000100000100000000000100100$$
$$0000010000010010001000001000000000000001100...]$$

Применим к данной последовательности разработанный нами алгоритм шифрования и исследуем его влияние на структуру данных после каждого выполненного раунда. Для анализа изменения свойств шифруемой последовательностями воспользуемся некоторыми, наиболее показательными тестами из [5]. Так, графический тест распределения на плоскости после выполнения первых трех раундов зашифрования изображен на рис. 3.

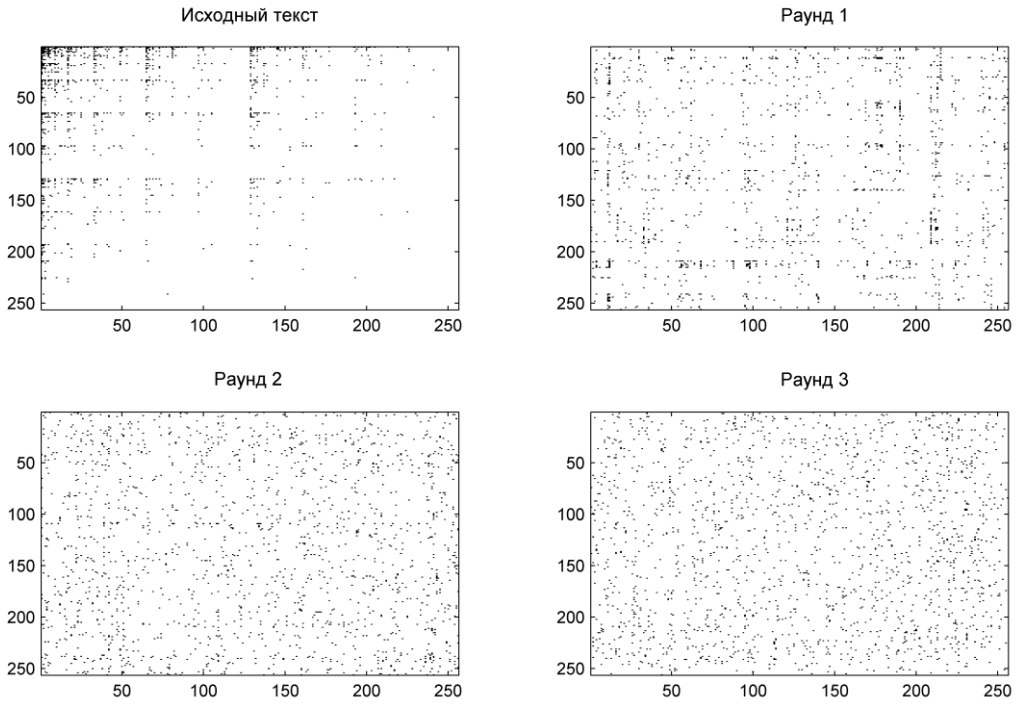

Рисунок 3. Распределение на плоскости для трех раундов

Как видно из анализа рис. 3 уже к третьему раунду шифрования можно говорить об устоявшемся случайном характере шифруемой информации, её хороших стохастических свойствах несмотря на детерминированную природу исходного текста и ключа. Ясно, что при выполнении последующих раундов криптопреобразования данное свойство будет только закрепляться.

Важным тестом эффективности работы криптографического преобразования является выполнение для криптограммы свойства k-граммного

распределения с большой точностью. На рис. 4 приведены графики 3-граммного распределения [5, 8] для первых трех рандов шифрования разработанного алгоритма.

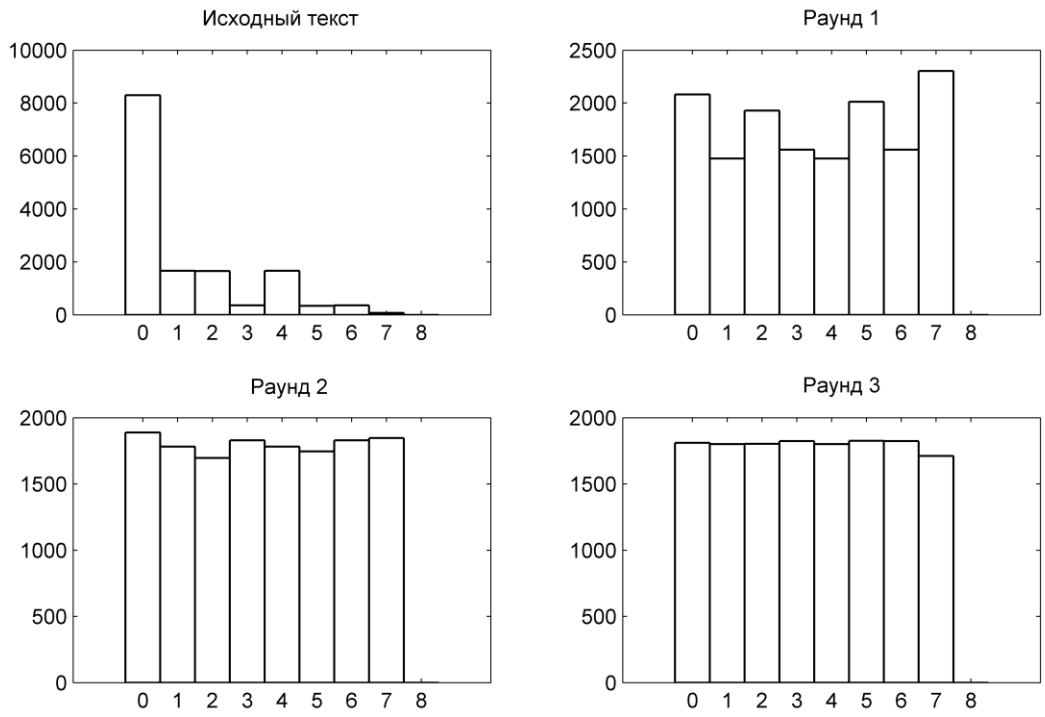

Рисунок 4. 3-граммное распределение для трех раундов

Таким образом, мы можем говорить что уже после второго раунда шифрования статистика исходного текста практически полностью разрушается.

Другим примером стохастического теста может является построение битовой автокорреляционной функции (АКФ) исходной последовательности и последовательностей на каждом раунде шифрования. Представим динамику изменения бокового максимума битовой АКФ по семи раундам шифрования

$$0.4539 \xrightarrow{1} 0.1567 \xrightarrow{2} 0.0439 \xrightarrow{3} 0.0331 \xrightarrow{4} 0.0342 \xrightarrow{5} 0.0297 \xrightarrow{6} 0.0314 \xrightarrow{7} 0.0356$$

Проведенные статистические тесты для других выбранных исходных текстов подтверждают высокие показатели диффузии и конфузии, а также и лавинного эффекта.

В заключении отметим основные результаты проведенных исследований: в статье разработан алгоритм шифрования данных на основе динамического

изменения размеров криптографических примитивов в различных раундах шифрования. Применение данного подхода позволяет улучшить эффективность преобразования информации в смысле реализации концепции диффузии и конфузии. Проведенные стохастические тесты зашифрованного текста после каждого раунда подтвердили высокую эффективность предложенного алгоритма, тогда как его число уровней защиты является достаточно большим и может быть легко масштабировано.

Таким образом, разработанный алгоритм может представлять интерес для практического применения.

Список литературы:

1. Blum, L. A Simple Unpredictable Pseudo-Random Number Generator / Lenore Blum, Manuel Blum, and Michael Shub. — *SIAM Journal on Computing*, 1986. — volume 15. — P. 364—383.
2. *Nyberg, K.* Differentially uniform mappings for cryptography. In Advances in cryptology / K. Nyberg // Proc. of EUROCRYPT'93. — Berlin, Heidelberg, New York. — 1994. — vol.765, Lecture Notes in Compuer Springer-Verlag. — P.55 — 65.
3. *Shannon C.* Programming a Computer for Playing Chess / C. Shannon. — Philosophical Magazine. — 1950. — Т. 7/41. — № 314. — С. 256—275.
4. *Жданов, О.Н.* Методика выбора ключевой информации для алгоритма блочного шифрования / О.Н. Жданов. — М.: ИНФРА-М, 2013 г.. — 90 с.
5. *Иванов, М.А.* Теория, применение и оценка качества генераторов псевдослучайных последовательностей / М.А. Иванов, И.В. Чугунков. — М.: КУДИЦ-ОБРАЗ, 2003. — 240 с.
6. *Мазурков М.И.* Криптографические свойства нелинейного преобразования шифра Rijndael на базе полных классов неприводимых полиномов / М.И. Мазурков, А.В. Соколов. — Праці Одеського політехнічного університету, 2012. — Вип. 2(39). — С.183—189.
7. *Мазурков, М.И.* Метод синтеза S-блоков по критерию нулевой корреляции между выходными и входными векторами данных и строгому лавинному критерию / М.И. Мазурков, А.В. Соколов // Известия высших учебных заведений. Радиоэлектроника. — 2014. — Т. 57, N 8. — С. 54—60.
8. *Мазурков, М.И.* Методы синтеза двоичных псевдослучайных последовательностей со свойством k-граммного распределения / М.И. Мазурков, А.В. Соколов // Пр. Одес. політехн.ун-ту. — Одесса, 2012. — Вып. 1 (38). — С. 188 — 198.
9. *Рябко, Б.Я.* Основы современной криптографии и стеганографии / Б.Я. Рябко, А.Н. Фионов. — М: Горячая линия — Телеком, 2010. — 232 с.
10. *Соколов, А.В.* Новые методы синтеза нелинейных преобразований современных шифров / А.В. Соколов. — Lap Lambert Academic Publishing, 2015. — 100 с.
11. *Чалкин, Т.А.* Разработка методики выбора параметров для алгоритма построения узлов замен блочного шифра ГОСТ 28147-89 /Т.А. Чалкин // Актуальные проблемы безопасности информационных технологий: материалы III Международной научно-практической конференции / под общей ред. О.Н. Жданова, В. В. Золотарева. — Сиб. гос. аэрокосмич. ун-т. — Красноярск, 2009. — С. 33—38.
12. *Чалкин Т.А.,* Жданов О.Н. Программный комплекс построения и тестирования ключевой информации для шифрования данных по алгоритму ГОСТ 28147-89. // Свидетельство о государственной регистрации программ для ЭВМ № 2011613877.