ISSN 1814-1196
Научный вестник НГТУ
том 76, № 3, 2019, с. 87–96

http://journals.nstu.ru/vestnik
Science Bulletin of the NSTU
Vol. 76, No. 3, 2019, pp. 87–96

ИНФОРМАТИКА,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И УПРАВЛЕНИЕ

INFORMATICS,
COMPPUTER ENGINEERING
AND MANAGEMENT

УДК 519.7

DOI: 10.17212/1814-1196-2019-3-87-96

Применение статистических тестов NIST для анализа выходных последовательностей блочных шифров*

А.А. ПЕРОВ

630099, РФ, г. Новосибирск, ул. Каменская, 56, Новосибирский государственный университет экономики и управления

perov_artem@inbox.ru

Современные итеративные блочные шифры являются одним из наиболее востребованных средств обеспечения защищенного обмена информацией в высокоскоростных сетях передачи данных. Широкое применение данной технологии и развитие вычислительных мощностей порождают целый перечень угроз криптоанализа шифров. Обеспечение криптографической стойкости является в данном случае одним из ключевых аспектов криптографического алгоритма, однако криптографическая стойкость невозможна без обеспечения удовлетворительных статистических свойств, так как ряд атак на блочные шифры основан как раз на статистических уязвимостях выходной последовательности. Дизайнеры используемых на сегодняшний день криптографических алгоритмов установили определенный запас при выборе характеристик работы шифров, тогда как выходная последовательность многих алгоритмов становится неотличима от случайной за меньшее, чем полное, число раундов шифрования. Некоторое сокращение такого параметра блочного шифра, как число раундов, позволит обеспечить удовлетворительные статистические свойства, при этом увеличив скорость работы алгоритма. Для проверки статистических свойств шифра проводится статистический анализ с помощью специализированных тестов, который, как правило, сопряжен с рядом сложностей.

В настоящей статье рассматривается задача анализа выходных последовательностей блочных шифров с целью поиска оптимального (минимального) числа раундов шифрования, при котором шифртекст неотличим от случайного. Изложены основные принципы работы статистических тестов. Описана технология обеспечения взаимодействия реализаций статистических тестов и криптоалгоритмов, которая предложена и реализована автором средствами языков программирования. Продемонстрированы новые, полученные в результате экспериментов характеристики работы для алгоритмов из библиотеки CPPCRYPTO.

Ключевые слова: криптография, итеративные криптоалгоритмы, блочные шифры, легковесные шифры, ключевая последовательность, раунд шифрования, статистические тесты, статистический анализ

* *Статья получена 30 декабря 2018 г.*

ВВЕДЕНИЕ

Современные итеративные криптосистемы помимо секретного ключа шифрования используют так называемые раундовые ключи. Принимая во внимание основное правило криптографии – принцип Керкгоффса [1], который гласит о том, что за конфиденциальность шифртекста отвечает только ключевая последовательность, можно сделать вывод о том, что задача генерации качественного ключа является одной из важных. Чтобы сделать вывод о работе генератора псевдослучайных чисел, а также оценить степень случайности выходной последовательности криптоалгоритма, необходимо провести тестирование статистических характеристик. Необходимость в генерации качественной псевдослучайной последовательности, а также в соблюдении хороших статистических свойств обусловлена тем, что ни шифртекст, ни ключевая последовательность не должны быть предсказуемы для злоумышленника. Имея информацию о статистических уязвимостях ключа шифрования, криптоаналитик может существенно снизить диапазон перебираемых ключей, что позволит реализовать атаку полным перебором. В ситуации, когда злоумышленник перехватывает зашифрованное сообщение, неудовлетворительные статистические свойства могут послужить причиной атаки по известному шифртексту.

Следует отметить, что задачи, которые решают криптографические алгоритмы, сводятся к получению последовательности из двоичных символов, а не байт. В данном случае качественный генератор можно сравнить с идеально ровной монетой, вероятность выпадения каждой из сторон которой строго равна 0,5. Генераторы случайных чисел можно поделить на два основных типа: истинно случайные физические генераторы/датчики случайных чисел и псевдослучайные программные датчики/генераторы случайных чисел. Первые принимают на вход некий случайный бесконечный процесс, а на выходе дают бесконечную (зависит от времени наблюдения) последовательность 0 и 1. Вторые представляют собой заданную разработчиком детерминированную функцию, которая инициализируется так называемым зерном, после чего также на выходе выдает последовательность 0 и 1. Зная это зерно, можно предсказать всю последовательность. Хороший программный датчик случайных чисел тот, для которого невозможно предсказать последующие значения, имея всю историю предыдущих значений, то есть не имея зерна. Задача восстановления предыдущего члена последовательности, заключающаяся в определении элемента последовательности ai-1 по известным k членам последовательности aiai+1ai+2...ai+k-1, называется непредсказуемостью влево. Существует и обратная задача предсказания следующего члена последовательности, при которой по известным k членам последовательности ai-k+1ai-k+2...ai-1ai предсказывается значение ai+1, она называется непредсказуемостью вправо [2].

Использование истинно случайных (физических) датчиков случайных чисел сопряжено с рядом проблем.

- Случайное явление/процесс, которое берется за основу, может быть не способно выдавать числа с нужной скоростью.
- Степень случайности некоторых физических явлений можно поставить под сомнение. Например, электромагнитный шум может быть суперпозицией нескольких более-менее однообразных периодических сигналов.

Ряд статистических тестов разработан NIST [3]. В основе каждого из этих тестов лежит задача вычисления статистики, характеризующей некое

свойство последовательности, после чего эта статистика сравнивается с эталонной статистикой, которую дает идеально случайная последовательность. Эталонная статистика выводится математически, чему посвящено множество теорем и научных трудов по криптографии и теории чисел. Тесты NIST в рамках исследований выходных последовательностей криптографических систем уже применялись в работе [4].

В основе тестов лежит понятие нулевой гипотезы. Нулевая гипотеза – это предположение, что между фактами появления чисел имеется какая-либо взаимосвязь. Иными словами, за нулевую гипотезу принимается предположение, что последовательность является истинно случайной (знаки которой появляются равновероятно и независимо друг от друга). Следовательно, если такая гипотеза верна, то генератор отвечает хорошим статистическим характеристикам.

Проверка гипотезы заключается в том, что имеется статистика, подсчитанная на основе собранных данных (например, по уже сгенерированным ключам). С другой стороны, имеется эталонная статистика, получаемая математическими методами (вычисленная сугубо теоретически), которая бы имела идеально случайную последовательность. Фактическая статистика, конечно, не может сравняться с эталонной: насколько бы ни был хорош генератор, он не может быть идеален. Для этого вводится некоторая доля погрешности (например, 0.05).

Таким образом, существует 4 итоговых результата.

- Сделан вывод о том, что последовательность случайна, и это верный вывод.
- Сделан вывод о том, что последовательность не случайна, хотя она была на самом деле случайна. Такие ошибки называют ошибками первого рода.
- Последовательность признана случайной, хотя на самом деле таковой не является, такие ошибки называют ошибками второго рода.
- Последовательность справедливо отбракована.

В каждом тесте вычисляется так называемое P-значение: это вероятность того, что генератор произведет последовательность не хуже, чем гипотетический истинный. Если P-значение = 1, то последовательность идеально случайна, а если она равно нулю, то последовательность полностью предсказуема. В дальнейшем P-значение сравнивается с α, и если оно больше α, то нулевая гипотеза принимается и последовательность признается случайной, в противном случае отбраковывается.

В тестах берется $\alpha = 0.01$. Из этого следует:

- если P-значение ≥ 0.01, то последовательность признается случайной с уровнем доверия 99 %;
- если P-значение < 0.01, то последовательность отбраковывается с уровнем доверия 99 %.

1. ПОСТАНОВКА ЗАДАЧИ

Для генерации псевдослучайных последовательностей из блочного шифра используется генератор псевдослучайных последовательностей, построенный на базе блочного шифра посредством использования режима шифрования CTR. Режим CTR (counter mode, режим счетчика) предполагает

возврат на вход соответствующего алгоритма блочного шифрования значения некоторого счетчика, накопленного с момента старта, и делает из блочного шифра потоковый, то есть генерирует последовательность, к которой применяется операция XOR с текстом сообщения. Исходный текст и блок зашифрованного текста имеют один и тот же размер блока, как и основной шифр.

Шифрование в режиме CTR можно представить следующей формулой:

$$C_i = P_i \oplus E_k(Ctr_i), \quad (1)$$

где C_i – i-й блок шифротекста; P_i – i-й блок открытого текста, $E_k(x)$ – функция шифрования блока x с ключом k; Ctr_i – значение счетчика для i-го блока;

В явном виде Ctr_i можно записать так:

$$Ctr_i = R[1]|N_i[1]\|R[2]|N_i[2]\|R[3]|N_i[3]\|R[4] \vee N_i[4], \quad (2)$$

где $N_i[j]$ – j-я пара байт числа номера i-го блока; $R[i]$ – i-я пара байт случайного числа R; $\vee$ – операция конкатенации.

Случайное число R определяется однократно перед началом работы алгоритма. В коде данный алгоритм реализован в функции generate_ctr, которая в качестве единственного аргумента принимает указатель на массив из четырех 32-битных чисел, в который будет записано значение счетчика для i-го блока. В данном случае переменная nonce – случайное 64-битное число, counter – номер блока.

```
u64 nonce = 0x1F3C091AD5B3CFAB;
u64 counter = 0x0000000000000000;

void generate_ctr(u32* result) {
    u16 nonce16[4];
    u16 counter16[4];
    _64to16(&nonce, nonce16, 1);
    _64to16(&counter, counter16, 1);

    for (int i = 0; i < 4; i++) {
        u16 temp[2] = {counter16[3 - i], nonce16[3 - i]};
        _16to32(temp, &(result[i]), 1);
    }

    if (counter == UINT64_MAX) {
        counter = 0;
    } else {
        counter++;
    }
}
```

Данная конструкция создана для того, чтобы унифицировать значение для всех шифров, так как у разных алгоритмов размер блока варьируется от 32 до 128 бит. Для эксперимента требуется, чтобы у каждого шифра в блоке была и часть счетчика, и часть случайного числа. Таким образом, для каждого шифра и для разного количества раундов генерируется псевдослучайная последовательность байт заданной длины, которую необходимо подвергнуть тестированию набором статистических тестов NIST.

2. СХЕМА ЭКСПЕРИМЕНТОВ

Для проведения тестов были выбраны алгоритмы, представленные в библиотеке CPPCRYPTO. Ранее автором были проделаны эксперименты над шифрами из библиотеки BLOC [5,6], которые проводились тестами «Стопка книг» [7] и «Адаптивный криптерий хи-квадрат» [8] для изучения вопроса борьбы с атакой-различителем. Эта универсальная атака относится ко всем симметричным криптоалгоритмам. Задача шифра преобразовывать информацию так, чтобы шифрованный текст выглядел как случайный, то есть вероятность появления каждого символа исходного алфавита была одинакова, однако современные шифры являются детерминированными алгоритмами, неспособными генерировать истинно случайные последовательности, поэтому теоретически для любого шифра эту «неслучайность» можно обнаружить. Таким образом, важным требованием к шифру является неотличимость шифрованного текста от случайной последовательности никакими известными тестами, методами, критериями. Если шифрованный текст удается отличить от равномерно распределенных случайных чисел, то это является интересным научным результатом и указывает на недостаток шифра. Более того, подобный недостаток может быть использован для определения секретного ключа [9].

Проведенные ранее исследования показали, что для выходной последовательности урезанных версий алгоритмов возможно обеспечение удовлетворительных статистических свойств, хотя и были проведены только двумя тестами. Для получення новых результатов тестирования шифров [10] были добавлены 15 статистических тестов NIST:

- частотный побитовый тест;
- частотный блочный тест;
- тест на последовательность одинаковых битов;
- тест на самую длинную последовательность единиц в блоке;
- тест рангов бинарных матриц;
- спектральный тест;
- тест на совпадение неперекрывающихся шаблонов;
- тест на совпадение перекрывающихся шаблонов;
- универсальный статистический тест Маурера;
- тест на линейную сложность;
- тест на периодичность;
- тест приблизительной энтропии;
- тест кумулятивных сумм;
- тест на произвольные отклонения;
- другой тест на произвольные отклонения.

В рамках существующей реализации тесты NIST работают на базе библиотеки GSL (GNU Scientific Library), которая написана на языке C и содержит значительное количество функций – от элементарных математических операций и операций с комплексными числами до численных методов дифференцирования, интерполяции, аппроксимации, решения дифференциальных уравнений, wavelet-преобразования и многих других. За счет широкого спектра математических функций объект исследования подвергается тщательному всестороннему анализу.

На вход алгоритму в качестве открытого текста подавался поток нулей длиной 2^{27} бит. Для работы интегрированных тестов статистических тестов были сгенерированы бинарные файлы с выходными последовательностями криптосистем в формате <имя шифра>_<количество раундов>_ciphertext.bin, тем самым создав структуру каталогов.

Корректное взаимодействие шифров и тестов обеспечил дополнительный скрипт, написанный на языке Python, задачей которого было рекурсивным способом обойти каталоги с шифртекстами и сформировать отчет о тестировании, запуская исполняемый файл тестов и подавая на вход сгенерированные файлы с выходными последовательностями. Ниже приводится фрагмент программного кода скрипта:

```
      For filepath in result: reports_folder =
filepath.parent.joinpath('reports')
                  if not reports_folder.exists():
                        reports_folder.mkdir()

                  input_file = str(filepath)
                  output_file = re-
ports_folder.joinpath(filepath.name.replace('.bin', '.txt'))

                  args = ("./statisticaltests", input_file,
output_file)  # запускаем тесты, указываем файл с
    p = subprocess.Popen(args, stdout=subprocess.PIPE)
            p.wait()
```

После прохождения тестов генерируется выходной файл с отчетом, отличающийся от исходного шифртекста расширением .txt. В файл отчета записываются полные данные о прошедшем тесте. Например, для шифра ktantan64 с тремя раундами фрагмент отчета выглядит следующим образом:

```
Frequency test FAILURE
Nb. of ones: 636280
Nb. of zeros: 363720
P-value: 0
Block frequency test FAILURE
Nb. of blocks: 50
Block length: 20000
Nb. of discarded bits: 0
P-value: 0
```

В данный отчет собираются все значения проведенных над шифртекстом тестов. Для однозначной трактовки результатов теста также формируются короткие отчеты с помощью вышеописанного скрипта, фрагмент которого приводится ниже.

```
result_types = ('SUCCESS', 'FAIL')
short_report_file = reports_folder.joinpath(filepath.name.replace('.bin', '_short.txt'))
    with output_file.open('r') as report, short_report_file.open('w') as short_report:
            for line in report.readlines():
                    line = str(line)
    if any(substr in line for substr in result_types):
            short_report.write(line)
```

Как видно из вышеприведенного кода, формирование отчета происходит посредством поиска результата success или failed и переносит в файл с коротким отчетом информацию о выполненном тестировании.

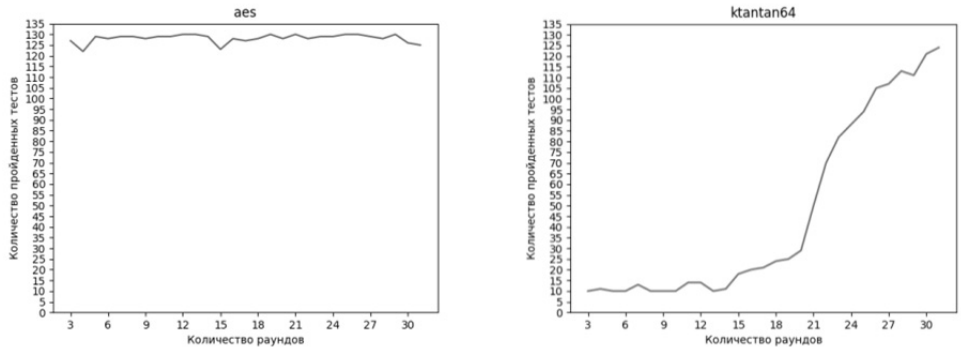
3. РЕЗУЛЬТАТЫ

В соответствии с критерием Пирсона вычислим значение хи-квадрат по формуле

$$x^2 = \sum_{i=1}^{r} \frac{(v_i - np_i)^2}{np_i},$$

где v – успешно пройденные тестирования, а np = 150 (15 тестов проводятся на 10 разных ключах шифрования). Экспериментально найдем, что при v_i = 126 значение χ^2 = 3.841, что соответствует уровню значимости a = 0.05. Из этого следует, что при 126 пройденных тестах вероятность корректного определения случайности выходной последовательности шифра равна 95 % [11].

Для иллюстрации результатов тестирования разработанный скрипт создает графики зависимости проведенных тестов от количества раундов, на которых тесты были пройдены. Алгоритмы продемонстрировали разные результаты (рисунок).



AES *ktantan64*

Графики зависимостей количества пройденных тестов от числа раундов шифрования для алгоритмов AES и ktantan64

Graphs of the dependencies of the number of tests conducted on the number of encryption rounds for the AES and ktantan64 algorithms

Результаты экспериментов доказали возможность использования усеченных характеристик алгоритмов для соблюдения абсолютной случайности выходной последовательности.

В таблице приведены значения раундов для протестированных алгоритмов библиотеки CPPCRYPTO.

Результаты тестирований (Rounds – полное число раундов шифра, R_{min} – найденное значение усеченного числа раундов, % R – процентное соотношение усеченного числа раундов к полному)

Test results (Rounds are the total number of rounds of the cipher, R_{min} is the found value of the truncated number of rounds, % R is the percentage of the truncated number of rounds to the full)

Шифр	Rounds	R_{min}	% R
Anubis	12	3	25
Aria	12	3	25
Cast6	48	3	6
Кузнечик	10	5	50
Mars	32	4	12
SM4	32	9	28
Threefish	72	3	4
Twofish	16	3	19

Полученные результаты свидетельствуют о частичной сопоставляемости с прошлыми исследованиями. В качестве участников эксперимента тестами NIST взяты прошедшие опытную проверку шифры-участники криптографических конкурсов, некоторые из них обеспечивают удовлетворительные статистические свойства уже на первых раундах шифрования.

ЗАКЛЮЧЕНИЕ

Предложенная в настоящей работе схема тестирования алгоритмов и разработанная для этого ее программная реализация позволяют сделать процесс тестирования итеративных криптоалгоритмов системным и вариативным, а также несколько упрощает интеграцию новых шифров. Эксперименты над каждым шифром проводились на возрастающем числе раундов с помощью всех предложенных тестов NIST. Подтверждена выдвинутая гипотеза об обеспечении удовлетворительных статистических свойств алгоритмов из библиотеки CPPCRYPTO за меньшее, чем полное, число раундов. Рассмотренные в настоящей работе алгоритмы показали высокие результаты уже на стартовых раундах шифрования.

В будущих работах планируется подробно описать порядок интеграции новых шифров в разработанную систему.

СПИСОК ЛИТЕРАТУРЫ

1. *Рябко Б.Я., Фионов А.Н.* Основы современной криптографии и стеганографии. – М.: Горячая линия-Телеком, 2010. – 232 с.

2. *Слеповичев И.И.* Генераторы псевдослучайных чисел: учебное пособие. – Саратов: СГУ, 2017. – 118 с.

3. A statistical test suite for random and pseudorandom number generators for cryptographic applications / A. Rukhin [et al.]. – Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2000. – 152 p. – (NIST special publication; 800-22).

4. *Ryabko B., Monarev V.* Using information theory approach to randomness testing // Journal of Statistical Planning and Inference. – 2005. – Vol. 133, N 1. – P. 95–110.

5. *Cazorla M., Marquet K., Minier M.* Survey and benchmark of lightweight block ciphers for wireless sensor networks // IACR Cryptology ePrint Archive. – Report 2013/295.

6. *Cazorla M., Marquet K., Minier M.* Survey and benchmark of lightweight block ciphers for wireless sensor networks // Proceedings of the 10th International Conference on Security and Cryptography, SECRYPT 2013, Reykjavík, Iceland, 29–31 July, 2013. – Reykjavík, 2013. – P. 543–548.

7. *Рябко Б.Я., Пестунов А.И.* «Стопка книг» как новый статистический тест для случайных чисел // Проблемы передачи информации. – 2004. – Т. 40, вып. 1. – С. 73–78.

8. *Рябко Б.Я., Стогниенко В.С., Шокин Ю.И.* Адаптивный критерий $\chi 2$ для различения близких гипотез при большом числе классов и его применение к некоторым задачам криптографии // Проблемы передачи информации. – 2003. – Т. 39, вып. 2. – С. 53–62.

9. *Пестунов А.И., Перов А.А., Пестунова Т.М.* О некоторых направлениях научных исследований в области криптоанализа симметричных алгоритмов // Вестник НГУЭУ. – 2016. – № 3. – С. 280–298.

10. *Перов А.А., Пестунов А.И.* Статистическое тестирование современных итеративных блочных шифров с помощью программной библиотеки «УНИБЛОКС-2015» // Инновации в жизнь. – 2016. – № 2. – С. 89–97.

11. *Ивченко Г.И., Медведев Ю.И.* Введение в математическую статистику: учебник. – М.: ЛКИ, 2010. – 600 с.

Перов Артём Андреевич, старший преподаватель кафедры информационных технологий Новосибирского государственного университета экономики и управления. Основное направление научных исследований – криптография. Имеет 14 публикаций. E-mail: perov_artem@inbox.ru

Perov Artem, a senior lecturer at the Novosibirsk State University of Economics and Management. The main research subject is cryptography. He is the author of 14 research papers. E-mail: perov_artem@inbox.ru

DOI: 10.17212/1814-1196-2019-3-87-96

Using NIST statistical tests for the analysis of the output sequences of block ciphers[*]

A.A. PEROV

Novosibirsk, State University of Economics and Management, 56 Kamenskaya Street, Novosibirsk, 630099, Russian Federation

perov_artem@inbox.ru

Abstract

Modern iterative block ciphers are one of the most popular methods for providing a secure information exchange in internet networks. A widespread use of this technology and the development of computing power give rise to a whole list of threats to cryptanalysis of ciphers. Ensuring cryptographic security is in this case one of the key aspects of the cryptographic algorithm, but cryptographic security is impossible without providing satisfactory statistical properties since a number of attacks on block ciphers are based on statistical vulnerabilities of the output sequence. The designers of the cryptographic algorithms used today have established a certain margin when choosing the characteristics of the cipher, while the output sequence of many algorithms becomes indistinguishable from the random sequence for less than the total number of encryption rounds. Some reduction of such a parameter of the block cipher as the number of rounds will provide satisfactory statistical properties while increasing the speed of the algorithm. To check the statistical properties of the cipher statistical analysis is carried out using specialized tests, which, as a rule, involve a number of difficulties.

This article describes the task of analyzing the output of block cipher sequences in order to find the optimal (minimum) number of encryption rounds, in which the cipher text is indistinguishable from the random one. The basic principles of the statistical tests are also described.

[*] *Received 30 December 2018.*

The technology of ensuring the interaction of implementations of statistical tests and cryptoalgorithms, which is proposed and implemented by the author by means of programming languages, is described. New, obtained as a result of experiments characteristics of work for algorithms from the CPPCRYPTO library are demonstrated.

Keywords: cryptography, iterative cryptographic algorithms, block ciphers, lightweight ciphers, key, encryption round, statistical tests, statistical analysis

REFERENCES

1. Ryabko B.Ya., Fionov A.N. *Osnovy sovremennoi kriptografii i steganografii* [Fundamentals of modern cryptography and steganography]. Moscow, Goryachaya liniya-Telekom Publ., 2010. 232 p.

2. Slepovichev I.I. *Generatory psevdosluchainykh chisel* [Pseudo random number generators]. Saratov, SGU Publ., 2017. 118 p.

3. Rukhin A.L. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Gaithersburg, MD, U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2000. 152 p. *NIST special publication*, 800-22.

4. Ryabko B., Monarev V. Using information theory approach to randomness testing. *Journal of Statistical Planning and Inference*, 2005, vol. 133, no. 1, pp. 95–110.

5. Cazorla M., Marquet K., Minier M. Survey and benchmark of lightweight block ciphers for wireless sensor networks. *IACR Cryptology ePrint Archive*. Report 2013/295.

6. Cazorla M., Marquet K., Minier M. Survey and benchmark of lightweight block ciphers for wireless sensor networks. *Proceedings of the 10th International Conference on Security and Cryptography, SECRYPT 2013*, Reykjavík, Iceland, 29–31 July, 2013, pp. 543–548.

7. Ryabko B.Ya., Pestunov A.I. "Stopka knig" kak novyi statisticheskii test dlya sluchainykh chisel ["Book Stack" as a new statistical test for random numbers]. *Problemy peredachi informatsii – Problems of Information Transmission*, 2004, vol. 40, iss. 1, pp. 73–78.

8. Ryabko B.Ya., Stognienko V.S., Shokin Yu.I. Adaptivnyi kriterii $\chi 2$ dlya razlicheniya blizkikh gipotez pri bol'shom chisle klassov i ego primenenie k nekotorym zadacham kriptografii [Adaptive $\chi 2$ test for discriminating between close hypotheses with a large number of classes and its application to some cryptography problems]. *Problemy peredachi informatsii – Problems of Information Transmission*, 2003, vol. 39, iss. 2, pp. 53–62.

9. Pestunov A.I., Perov A.A., Pestunova T.M. O nekotorykh napravleniyakh nauchnykh issledovanii v oblasti kriptoanaliza simmetrichnykh algoritmov [On some scientific problems in cryptanalysis of symmetric algorithms]. *Vestnik NGUEU – Vestnik NSUEM*, 2016, no. 3, pp. 280–298.

10. Perov A.A., Pestunov A.I. Statisticheskoe testirovanie sovremennykh iterativnykh blochnykh shifrov s pomoshch'yu programmnoi biblioteki "UNIBLOKS-2015" [Statistical testing of modern iterative block codes by means of program library "UNIBLOKS-2015"]. *Innovatsii v zhizn' – Innovations in Life*, 2016, no. 2, pp. 89–97.

11. Ivchenko G.I., Medvedev Yu.I. *Vvedenie v matematicheskuyu* statistiku [Introduction to mathematical statistics]. Moscow, LKI Publ., 2010. 600 p.

Для цитирования:

Перов А.А. Применение статистических тестов NIST для анализа выходных последовательностей блочных шифров // Научный вестник НГТУ. – 2019. – № 3 (76). – С. 87–96. – DOI: 10.17212/1814-1196-2019-3-87-96.

For citation:

Perov A.A. Primenenie statisticheskikh testov NIST dlya analiza vykhodnykh posledovatel'nostei blochnykh shifrov [Using NIST statistical tests for the analysis of the output sequences of block ciphers]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2019, no. 3 (76), pp. 87–96. DOI: 10.17212/1814-1196-2019-3-87-96.