УДК 621.391:519.72

КАЗАКОВА Н.Ф., ЩЕРБИНА Ю.В.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ ПРЕДВАРИТЕЛЬНОГО АНАЛИЗА КРИПТОГРАФИЧЕСКИХ ПРОГРАММНЫХ ГЕНЕРАТОРОВ ПСП

Аннотация. *Проведен анализ состояния развития инструментальных средств для тестирования псевдослучайных последовательностей, применяемых в криптографии и особенности применения критерия согласия* χ^2.

Annotation. *The analysis of development of tools status is conducted for testing of pseudocasual sequences, applied in cryptography and feature of application the criterion of consent* χ^2.

Постановка проблемы в общем виде и ее связь с научными и практическими задачами. В настоящее время, в области защиты конфиденциальности передаваемой в информационно-телекоммуникационных системах информации, наблюдается тенденция повышения интереса к разработке потоковых шифров. Такие шифры, в отличие от блочных шифров, хотя и несколько уступают им в криптографической стойкости, обладают более высокой производительностью, что позволяет их использовать в реальном масштабе времени. Именно в силу этих причин, европейским криптологическим сообществом ECRYPT был объявлен открытый конкурс (2004÷2008г.г.) на разработку новых потоковых шифров – eSTREAM (*ECRYPT Stream Cipher Project*) [1] с целью выявления наиболее достойного соискателя на использование в качестве стандарта для стран европейского сообщества.

Разработка подобных шифров сводится к построению генераторов шифрующей гаммы, максимально приближающейся по своим статистическим свойствам к случайным последовательностям с равномерным законом распределения вероятностей формируемых символов. Алгоритмы формирования псевдослучайных последовательностей (ПСП) с высокой степенью «случайности» находят широкое применение и при создании защищенных криптографических протоколов в качестве формирователей ключевой информации.

Хотя задача построения равномерно распределенных псевдослучайных последовательностей в своей постановке проста, в действительности ее решение связано с целым рядом проблем, требующих проведения серьезных научных исследований и глубокого владения математическим аппаратом в области статистики. И то и другое не может не вызывать серьезных трудностей у инженеров, занимающихся практической реализацией функций защиты информационных процессов.

Анализ научной и технической литературы (например, [2…4]) показывает, что к настоящему времени разработано достаточно большое количество инструментальных средств, позволяющих осуществлять предварительный анализ пригодности последовательностей, порождаемых генераторами ПСП, для нужд криптографии. Эти пакеты прикладных программ реализуют наборы тестов, призванные дать ответ на вопрос, возможно ли, зная некоторый участок формируемой гаммы, предсказать последующий (или предыдущий) ее символ с вероятностью, отличной от 1/2? При отрицательном ответе формируемая гамма признается действительно случайной. Программное обеспечение и описание наборов этих тестов является общедоступным и может быть получено через Интернет-сайты своих разработчиков.

Проблема, однако, состоит в том, что использование предлагаемого программного обеспечения предполагает глубокий комплексный анализ готового продукта. Прилагаемые описания этих тестов содержат комплексные программы испытаний датчиков ПСП и процедуры вычисления обобщенного показателя качества, учитывающего результаты от всех тестов, входящих в состав пакета. Эти тесты позволяют выявлять различные виды аномалий в псевдослучайной последовательности, которые в принципе могут быть использованы как уязвимости криптоалгоритма для организации на их основе атаки со стороны злоумышленников.

В то же время, инженерам-разработчикам необходимы инструментальные средства, реализующие простые и надежные процедуры тестирования на начальных и промежуточных этапах создания генераторов, позволяющих убедится в правильности выбранного пути. По

этой причине разработчики алгоритмов формирования ПСП, как правило, зачастую предлагают собственные средства испытания, подтверждающие качество созданного программного продукта и далеко не всегда используют комплекты тестов, рекомендованные NIST [2] или другими авторитетными организациями. Отчасти это объясняется еще и тем, что разработчику важен не столько обобщенный показатель «случайности» сформированной датчиком гаммы, сколько вид отдельной аномалии в ее составе, приводящей к ухудшению этой самой «случайности». По этой причине в каждом конкретном случае приходится искать свой подходящий способ тестирования, и это обстоятельство объясняет повышенное внимание многих исследователей этой области к построению новых тестов. В подтверждение этого свидетельствует тот факт, что например, для шифра RC4, созданного в 1987 году, и в котором, в последствии, были обнаружены некоторые отклонения от случайности, никто не мог предложить практически реализуемого подходящего теста в течение многих лет, несмотря на большое число работ, посвященных этому вопросу. Все это говорит о том, поиск новых эффективных тестов, пригодных для использования в качестве инструментальных средств в процессе проектирования остается пока ***актуальной и не решенной до конца задачей***.

Впервые практическое решение проблемы тестирования датчиков случайных чисел было предложено Дональдом Кнутом в его классической работе «Искусство программирования для ЭВМ» [5]. В ней рассматривается несколько тестов, позволяющих определить, насколько соответствует распределение вероятностей некоторой случайной величины в наблюдаемом процессе ожидаемому виду распределения.

Среди разработанных Д. Кнутом тестов, наиболее подходящим для нужд криптографии представляется тест, основанный на критерии согласия χ^2 Пирсона. В настоящее время он применяется для вычисления значения показателя качества в подавляющем большинстве тестов, входящих в состав криптографических пакетов. Ценность этого критерия определяется тем, что с его помощью можно напрямую оценить степень равномерности распределения вероятностей чисел, получаемых на выходе генератора ПСП, и это делает его пригодным для предварительной оценки не только самого генератора, но входящих в его состав отдельных компонентов. Упомянутая работа Д. Кнута была написана во времена становления вычислительной техники и не ориентирована на нужды криптографии. Основной упор автор делал на экономии ограниченных, по тем временам, вычислительных ресурсов и не ставил вопрос о точности метода, которая так важна в области криптоанализа.

Целью статьи является акцентирование внимания на особенностях применения критерия согласия χ^2 для подтверждения соответствия распределения вероятностей символов, формируемой псевдослучайной последовательности равномерному закону.

О применении χ^2-критерия написано много и он, без сомнения, является одним из наиболее часто применяемых. Однако, в большинстве литературных источников из области статистики, процедура его применения рассматривается в общей постановке. Между тем интерес к нему растет, и не только в связи с решением криптографических задач. В России, например, даже был выпущен в 2001 году стандарт, определяющий порядок его использования [7].

Рассмотрим суть и особенности решаемой задачи. Предположим, некоторый генератор порождает буквы алфавита $A = \{a_1, a_2, \ldots, a_S\}$, $S > 1$. Причем эти буквы представлены в вычислительной системе двоичными кодовыми комбинациями фиксированной длины m. Тогда общее число символов на выходе источника будет равно 2^m. При условии равномерности распределения их вероятностей будет выполняться простая гипотеза H_o, в соответствии с которой все символы равновероятны

$$H_o : p(a_1) = p(a_2) = \ldots = p(a_S) = 1/S.$$

Соответственно, предполагается, что альтернативная гипотеза H_1 состоит в отрицании этого утверждения.

Для подтверждения справедливости гипотезы формируется выборка $x_1, x_2, \ldots, x_n$, по которой определяются оценки вероятностей символов на выходе генератора. Величина n, как

показано в работе [5], должна быть такой, чтобы каждый символ, имел возможность быть сформированным хотя бы пять раз, иными словами, должно выполняться условие $np_i \geq 5$. Обычно эту величину n_i выбирают равной $5 \div 10$ и подсчитывают оценки вероятностей как $p_i^* = n_i / n$. Отсюда величина n должна удовлетворять условию $n \geq 5 * 2^m$. Так, например, если генератор формирует на выходе восьмибитовые символы, то их общее число будет равно 256 (от 0 до 255). Подсчитав величину n_i для каждого из них, можно определить показатель

$$\chi^2 = \sum_{i=1}^{2^m} \frac{(p_i^* - n/S)^2}{n/S},$$

который характеризует степень приближения реального распределения чисел на выходе генератора к равномерному закону.

Данный критерий основан на том, что с ростом величины n, показатель χ^2 сходится к распределению χ^2 с $(S-1)$ степенями свободы. Здесь S можно рассматривать как число интервалов, на которых наблюдается исследуемая величина, а единица определяет число параметров, вычисляемых на основе наблюдаемой статистики [7].

Для принятия решения относительно справедливости нулевой гипотезы H_o, следует определить уровень значимости $(1-\alpha)$. Здесь α – есть вероятность ошибки первого рода, означающая вероятность отвергнуть гипотезу H_o, когда она в действительности справедлива.

С учетом характера распределения величины χ^2, ее пороговое значение, соответствующее принятому уровню значимости $\chi^2_{(1-\alpha),(S-1)}$ может быть определено с помощью функции **chi2inv**$((1-\alpha),(S-1))$ пакета MATLAB. Например, для рассмотренного выше случая, величина $\chi^2_{0.99,255} = 311.5603$. Если этот порог величиной χ^2 превышен, то нулевая гипотеза отвергается. Например, при уровне $\alpha = 0{,}01$, в случае тестирования ста последовательностей, полученных от генератора случайных чисел с разными значениями ключа, не более чем для одной из них показатель χ^2 может превысить значение $\chi^2_{(S-1),(1-\alpha)}$.

В работе Дж.Тейлора [7] показано, что если расчетное значение показателя усреднять по большому числу испытуемых последовательностей, то расчетная величина $\chi^2_{ср}$, будет удовлетворять условию

$$\chi^2_{ср} \leq S - 1. \qquad (1)$$

Таким образом, если доля тестируемых последовательностей, которые успешно проходят этот тест не менее величины уровня значимости $(1-\alpha)$ и выполняется условие (1), можно считать, что испытуемый генератор прошел предварительную проверку и может быть испытан с помощью одного из прикладных тестовых пакетов [2…4]. Если же тест на основе критерия не дал положительных результатов, дальнейшие проверки бессмысленны.

Одним из «узких» мест этого теста является выбор количества символов выходного алфавита S или, иначе говоря, выбор числа интервалов, на которых наблюдаются символы выходной последовательности тестируемого генератора. В работе [8], утверждается, что число интервалов должно выбираться таким образом, чтобы на каждый символ выходного алфавита был свой интервал. То есть, если, например, генератор оперирует 32-х битными числами, то и число интервалов должно быть равно 2^{32}. Это утверждение делается на том основании, что в противном случае проявляется «двуличность процесса». Смысл этого негативного явления состоит в том, что если тестировать некоторую регулярную (или близкую к регулярной) последовательность, то она будет идентифицироваться как совершенно «случайная» с равномерным законом распределения вероятностей. И, чем длиннее размер слова в алфавите, тем меньше степень такой опасности.

Далее авторы этой работы разумно ставят вопрос о том, что при таком размере алфавита потребуется невероятно большой размер памяти и временных ресурсов для реализации теста.

С целью преодоления этой проблемы предлагается тест на основе так называемого алгоритма «Стопка книг». Его суть сводится к тому, что символы алфавита нумеруются в естественном порядке и при формировании датчиком случайного числа, символ с соответствующим номером перемещается на первое место в алфавите. При этом все символы, занимающие предыдущие позиции (слева), перемещаются вправо на один шаг. Это происходит аналогично тому, как извлеченная наугад книга из вертикальной стопки кладется на верхнее место, а все вышестоящие, опускаются вниз на одну позицию. Затем, весь диапазон чисел разбивается на относительно небольшое количество интервалов и дальше, как обычно, применяется критерий χ^2. Описанный алгоритм можно проиллюстрировать следующим рисунком.

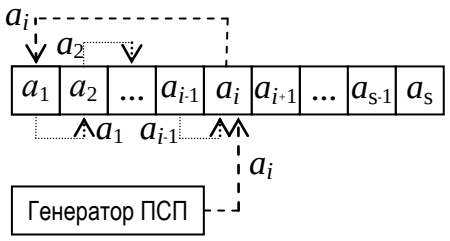

Рисунок 1 – Принцип перестановки символов алфавита
по методу «Стопка книг»

Справедливо делается утверждение, что при равновероятных символах вероятности попадания сформированного датчиком числа в любой из интервалов будут равны, а в противном случае более часто встречающиеся символы будут чаще оказываться в верхней части «стопки». На этом основании делается утверждение о большей чувствительности теста к неравномерности тестируемой последовательности, что и доказывается на высоком математическом уровне в целом ряде опубликованных авторами этого теста статей.

Не оспаривая в принципе утверждений, сделанных авторами этой работы, следует заметить, что количество интервалов разбиения (или размер символа m в алфавите) можно выбрать небольшим и в обычном, рассмотренном выше, χ^2-тесте. При этом возможная недостаточная «случайность» последовательности будет выявлена иными, не требующими значительных вычислительных ресурсов, дополнительными тестами при окончательных испытаниях генератора. Второе замечание является более серьезным и неприятным. Программная реализация теста «Стопка книг» потребует моделирования процедуры перемещения чисел в алфавите («в стопке»), а это, в свою очередь, также потребует больших вычислительных ресурсов, сводя декларируемые преимущества метода к минимуму.

Авторами данной стать были проведены сравнительные испытания обычного метода χ^2 и метода «Стопка книг» при равном количестве интервалов, для обоих методов. В качестве генератора ПСП использовался алгоритм RC4 с длиной блока в один байт ($m = 8$). На рисунке представлен график, отображающий результаты тестирования 100 последовательностей, длина каждой из которых составила 8192 байт. Здесь кривая 1 отображает величину показателя для обычного χ^2-теста, а кривая 2 – отображает усреднение этой величины с изменением числа тестов от 1-го до 100. Соответственно, кривая 3 отображает величину показателя для теста «Стопка книг», а кривая 4 – отображает усреднение этой величины с ростом числа тестов от 1-го до 100.

Сравнение этих характеристик показывает близость результатов, которые дают оба теста при равных условиях для испытуемых последовательностей и пригодны для предварительного анализа генераторов, разрабатываемых для криптографических нужд.

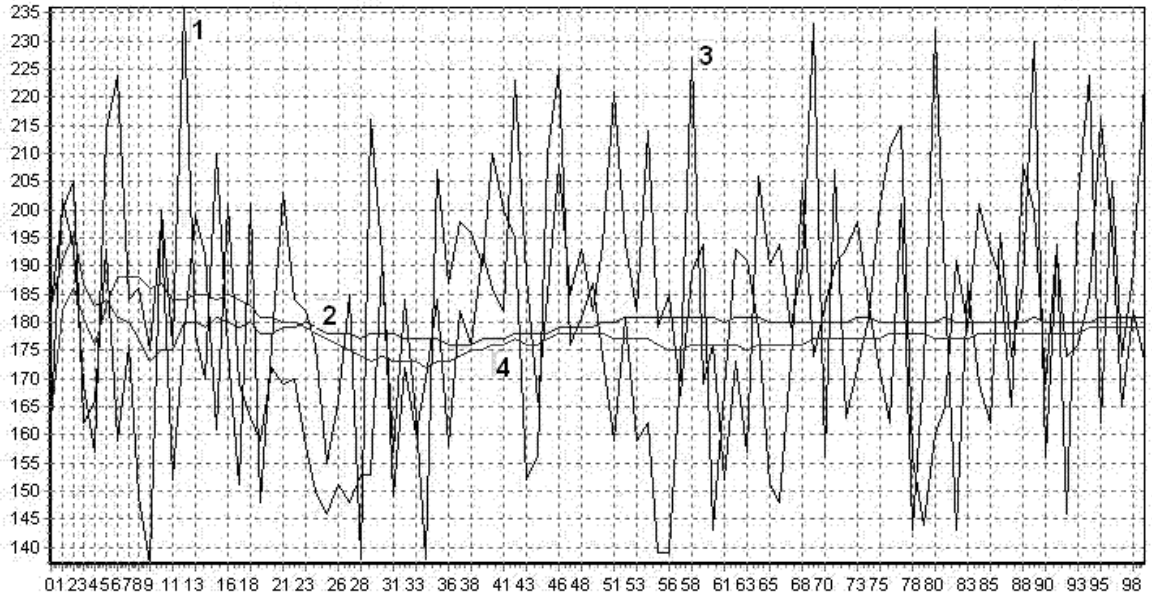

Рисунок 2 – Изменение показателей результатов тестирования

В ***заключение*** отметим, что, как указано в руководстве к пакету тестов, разработанных NIST [2], тестирование, не зависимо от того, проводится ли оно отдельными тестами, или пакетами тестов, не является частью криптоанализа. По его результатам делается предварительный анализ стойкости криптографического генератора. Иное дело, что общеизвестные пакеты, прошедшие испытание временем, позволяют провести комплексную проверку генератора для выявления тех аномальных мест в равномерно распределенной последовательности, которую не обнаруживают традиционные статистические методы.

Учитывая, что новые шифры разрабатываются с учетом уязвимостей, ставших причинной удачных криптографических атак на уже известные алгоритмы, есть основания полагать, что поиск новых методов тестирования будет продолжен.

СПИСОК ЛИТЕРАТУРЫ

1. eSTREAM, the ECRYPT Stream Cipher Project // [Электронный ресурс]: http://www.ecrypt.eu.org/stream/index.html
2. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. NIST Special Publication 800-22. May 15, 2001.
3. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness // [Электронный ресурс]: http://www.stat.fsu.edu/pub/diehard
4. Statistical test suite Crypt-X // [Электронный ресурс]: http://www.isi.qut.edu.au/resources/cryptx/
5. Кнут Д. Искусство программирования для ЭВМ. Т.2. – М.: Мир, 1977. – 727 с.
6. Правила проверки согласия опытного распределения с теоретическим. Часть I. Критерии типа хи-квадрат 14.12.2001 // [Электронный ресурс]: http://www.tcnti.ru/shop/catalog/index.php?docum=25096/
7. Тейлор Дж. Введение в теорию ошибок. Пер. с англ. – М.: Мир, 1985. – 272 с.
8. Дорошенко С.А., Лубкин А.М., Рябко Б.Я., Фионов А.Н. Экспериментальный анализ шифра RC4 и потоковых шифров, выдвинутых на конкурс ESTREAM. – Новосибирск: Сибирский гос. ун-т телекоммуникаций и информатики // [Электронный ресурс]: http://www.contrterror.tsure.ru/site/magazine8/05-14-Doroshenko.htm