УДК 004.05

Прокофьев А.О., Чугунков И.В., Матрюхина Е.А., Гриднева Е.А.

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский ядерный университет «МИФИ», г. Москва, Россия

ВОПРОСЫ ПОСТРОЕНИЯ ПРОГРАММНЫХ СИСТЕМ ОЦЕНКИ КАЧЕСТВА СТОХАСТИЧЕСКИХ АЛГОРИТМОВ*

АННОТАЦИЯ

В статье обсуждаются вопросы исследования свойств генераторов псевдослучайных чисел (ГПСЧ). Определяется роль ГПСЧ при построении компьютерных систем, в том числе защищенных. Перечисляются наиболее значимые задачи, решаемые с использованием генераторов. Приводятся требования, предъявляемые к качественным ГПСЧ, обсуждаются проблемы, возникающие при оценке их непредсказуемости. Рассматривается один из вариантов решения данных проблем, заключающийся в сведении задачи построения криптостойкого ГПСЧ к задаче построения статистически безопасного генератора. Определяются требования к статистически безопасному ГПСЧ. Показана необходимость комплексного исследования статистических свойств формируемых ГПСЧ последовательностей. Анализируются существующие наборы статистических тестов, выявляются их недостатки, не позволяющие осуществлять полнофункциональное тестирование псевдослучайных последовательностей. В частности, подвергается критике методика оценки результатов прохождения тестов, не обладающая должной гибкостью в вопросах совместного использования частных оценок конкретного теста и совокупной оценки результатов прохождения набора тестов. Формулируются требования к качественной системе оценки статистической безопасности ГПСЧ. Показана необходимость использования графических тестов наряду с оценочными. Предлагается структура полнофункциональной системы оценки статистической безопасности ГПСЧ и криптоалгоритмов, определяется назначение входящих в нее модулей, приводятся примеры работы системы. Также рассматриваются возможные пути повышения эффективности оценочных тестов, в частности, обсуждаются вопросы сокращения объемов необходимой для реализации теста памяти, а также возможность повышения быстродействия тестов и одновременного тестирования нескольких последовательностей за счет использования гибридных технологий (совместного использования вычислительных ресурсов центрального процессора и ядер графических видеокарт).

КЛЮЧЕВЫЕ СЛОВА

Генератор псевдослучайных чисел, статистические тесты, система оценки качества.

Anton Prokofiev, Ilya Chugunkov, Elena Matriukhina, Ekaterina Gridneva

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia

THE ISSUES OF BUILDING SOFTWARE SYSTEMS QUALITY EVALUATION OF STOCHASTIC ALGORITHMS

ABSTRACT

The paper is devoted to quality assessment of pseudo-random generators (PRNG). The PRNG role in computer system construction, including secure, is defined. The most significant problems solved with use of generators are listed. Requirements to qualitative PRNG are provided, its unpredictability assessment problems are discussed. One of problem solutions consisting in

* **Труды XI Международной научно-практической конференции «Современные информационные технологии и ИТ-образование» (SITITO'2016), Москва, Россия, 25-26 ноября, 2016**

reducing the problem of cryptographically strong PRNG constructing to the task of statistically secure generator constructing is considered. Requirements to statistically secure PRNG are defined. Need of complex statistical properties assessment of sequences formed by PRNG are shown. Sets of existing statistical tests are analyzed, its disadvantages which aren't allows to carry out full-function pseudorandom sequences testing are revealed. In particular, the passing tests estimation method without adequate flexibility in question of concrete test private estimate and tests set passing results cumulative assessment sharing are criticized. Requirements to PRNG's quality assessment systems are formulated. The need of graph test using along with estimated is shown. The structure of full-function PRNG statistical safety assessment system is offered, the assignment of modules entered it is defined and examples of system work are proposed. Possible ways to improve assessment test are also considered. In particular, the decreasing of memory volumes needed for the test implementation as well as the possibility of test speed increasing and several sequences simultaneous testing due to hybrid technology using (sharing of central processor computing resources and graphic video cards kernels) are discussed.

KEYWORDS

Pseudo-random number generators; statistical tests; quality assessment system.

Одной из основных характеристик любой компьютерной системы, независимо от ее уровня сложности и функционального назначения, является безопасность процессов сбора, хранения, обработки и передачи информации. Важнейшую роль в реализации данных процессов играют генераторы псевдослучайных чисел (ГПСЧ), которые используются для решения следующих задач [1]:

- автономного и встроенного диагностирования компонентов компьютерных систем;
- контроля хода выполнения программ с использованием сторожевых процессоров;
- внесения неопределенности в работу программных средств для обеспечения устойчивости к случайным и умышленным деструктивным воздействиям;
- помехоустойчивого кодирования, обеспечивающего обнаружение и исправление ошибок, возникающих при передаче данных по каналам связи [2];
- обеспечения аутентичности (целостности, достоверности, юридической значимости и пр.) информации;
- обеспечения секретности информации.

В связи со значимостью ГПСЧ к формируемым ими последовательностям предъявляются жесткие требования. Качественный ГПСЧ, ориентированный на использование в компьютерных системах ответственного назначения, должен удовлетворят следующим требованиям:

- непредсказуемость; аналитик, знающий принцип работы такого генератора, имеющий возможность исследовать фрагмент последовательности конечной длины, но не знающий ключевой информации, для предсказания следующего элемента последовательности не может предложить лучшей стратегии, чем бросание жребия;
- хорошие статистические свойства; формируемая генератором последовательность по своим статистическим свойствам должна быть неотличима от истинно случайной последовательности;
- большой период формируемой последовательности; например, при преобразовании информации для каждого элемента исходной последовательности необходимо использовать свой элемент псевдослучайной гаммы; малый период может привести к корреляции между выборками;
- эффективная программная и аппаратная реализация.

Непредсказуемость криптографических ГПСЧ основывается на недоказуемом предположении, что аналитику не хватит ресурсов (вычислительных, временных или стоимостных), для того чтобы инвертировать функцию обратной связи или функцию выхода генератора. Теория сложности вычислений, к сожалению, не в состоянии дать нижнюю границу сложности анализируемого алгоритма инвертирования. Иначе говоря, никогда нет гарантии, что алгоритм, сложность которого анализируется, самый эффективный из всех возможных.

Одним из вариантов решения вышеуказанной проблемы является сведение задачи построения криптостойкого ГПСЧ к задаче построения статистически безопасного генератора. Статистически безопасный генератор должен удовлетворять следующим требованиям:

- отсутствие в формируемой ГПСЧ последовательности каких-либо статистических

закономерностей, иными словами, ее неотличимость от истинно случайной;

- отсутствие корреляции между формируемыми ГПСЧ последовательностями.

Таким образом, возникает актуальная задача разработки методов и средств оценки статистической безопасности стохастических алгоритмов, то есть алгоритмов, основанных на использовании ГПСЧ. Более того, данный подход можно применять и к оценке качества криптографических алгоритмов. Не случайно при проведении конкурса AES (Advanced Encryption Standard), победителем которого стал Rijndael [4] – де-факто являющийся в настоящий момент международным стандартом шифрования, для оценки качества алгоритмов-участников активно использовались результаты их статистического тестирования.

Можно выделить следующие активно использующиеся системы оценки статистических свойств ГПСЧ:

- A statistical test suite for random and pseudorandom number generators for cryptographic applications Национального института стандартов и технологии США (Руководство NIST) [5];
- система DIEHARD Дж. Марсалья [6];
- подборка тестов Д. Кнута [7];
- система CRYPT-X [8].

Однако, для полнофункционального исследования ГПСЧ данных систем недостаточно. Связано это с тем, что все вышеупомянутые системы имеют жестко заданные параметры тестирования. В результате, система может просто не поддерживать анализ определенного типа неслучайности, требуемый пользователю, или поддерживать, но не применительно к заданным параметрам генератора.

Еще одной проблемой существующих систем является ориентированность последних на генераторы определенного вида. К примеру, DIEHARD предназначен для исследования 32-разрядных конгруэнтных генераторов, и большинство его тестов рассматривают наборы, состоящие именно из 32-разрядных чисел. Руководство НИСТ требует, чтобы исследуемая последовательность была битовой. Предположим, необходимо оценить статистические свойства 8-разрядного генератора. Как в этом случае формировать последовательность для тестирования? Ни одна из существующих систем никаких рекомендаций на этот счет не дает. В результате, при тестировании системой DIEHARD или Руководством НИСТ могут быть упущены некоторые особенности, присущие 8-разрядным генераторам и выявляемые только тестами, ориентированными именно на 8-разрядные последовательности. Разрабатывать же генераторы, формирующие последовательности, подходящие для той или иной системы, также неправильно, так как генератор предназначен для решения других задач, и это система тестирования должна подстраиваться под структуру генератора, а не наоборот.

Таким образом, процесс создания средства для исследования ГПСЧ является чрезвычайно сложным и требует учета разного рода нюансов и факторов. Можно сформулировать следующие требования к системе оценки качества ГПСЧ, позволяющей ей быть не только эффективной, но и, насколько это возможно, универсальной:

- наличие эмпирических и теоретических критериев. Процесс тестирования последовательностей занимает значительное время, особенно в случае последовательностей большой длины. Теоретические критерии позволяют дать заключения о свойствах генератора еще до тестирования последовательностей, следовательно, их наличие позволит сразу отбраковать генераторы с параметрами, заведомо приводящими к неудовлетворительным результатам. К сожалению, теоретические критерии ориентированы на генераторы конкретного типа, а иногда даже и на конкретные параметры данных генераторов, и поэтому могут быть непригодны для тестирования остальных генераторов. Собственно говоря, именно этим фактом и объясняется необходимость создания собственной системы для тестирования нового генератора. Одним из вариантов универсализации системы является создание и пополнение базы данных теоретических критериев, а затем обмен полученными данными с другими пользователями. Идеальным вариантом было бы создание сетевой или распределенной базы данных теоретических критериев, пополняемой пользователями. При наличии двух типов критериев пользователю, наряду с непосредственно последовательностью, необходимо будет ввести параметры генератора, ее сформировавшего. Сначала данные параметры обрабатываются при помощи теоретических критериев, отфильтровывая тем самым заведомо некачественные последовательности; оставшиеся после фильтрации последовательности обрабатываются при помощи эмпирических критериев;

- наличие оценочных и графических тестов. И те, и другие тесты играют важную роль в исследовании статистических свойств последовательности. Оценочные тесты возвращают результат своей работы в виде численной статистики, позволяя тем самым автоматизировать процесс тестирования. Кроме того, данные значения можно сопоставить с аналогичными показателями для других последовательностей, проведя тем самым сравнительный анализ генераторов. Графические же тесты позволяют увидеть общую картину, а кроме того, как показано в ряде работ [1; 9], позволяют выявить статистические слабости, не обнаруживаемые статистическими тестами. Также графические тесты могут использоваться для оценки изменений, вносимых нелинейными преобразованиями, а также для оценки автокорреляции и корреляции между последовательностями;
- градация тестов по их силе. При анализе последовательности может использоваться несколько десятков тестов. Выполнение каждого теста требует некоторого времени, которое для некоторых тестов (например, «Сжатие при помощи Алгоритма Лемпела-Зива») может быть значительным. В ряде случае для получения заключения о свойствах генератора необходимо провести исследование сотен или тысяч последовательностей. Таким образом, время до получения предварительных итогов исследования может быть значительным. Данное время можно сократить, если ввести градацию тестов по их силе. Более того, можно вести категории для тестов, например «быстрый», «медленный», «обязательный», «вспомогательный». Это позволит ввести политику тестирования. Например, вначале выполняются «быстрые обязательные» тесты, то есть тесты, непрохождение которых однозначно свидетельствует о непрохождении других тестов (например, если не пройден «частотный тест», то не будут пройдены «проверка серий», «проверка равномерности» и так далее). При успешном прохождении «быстрых обязательных» тестов будут выполняться тесты следующей категории и так вплоть до самых сложных специфических тестов. Каждую категорию тестов также можно проранжировать, останавливая тестирование в случае неудачных результатов для данной категории или реализую ветвящееся тестирование, когда выбор следующей категории зависит от результатов выполнения текущей;
- наличие средств оценки периода тестируемой последовательности. Возможны ситуации, когда в последовательности заданной длины содержится несколько периодов, предпериоды, вырожденные циклы и так далее. Для генераторов, ориентированных на использование в системах ответственного назначения, крайне нежелательны повторения в гамме, поэтому необходимо отслеживать, чтобы в формируемой последовательности не было периодичности. Кроме того, в случае периодичности тест работает вхолостую – ведь он фактически дублирует тут статистику, которую уже собрал в предыдущем периоде. Поэтому основное, с чего нужно начать исследование последовательности – это определение периода. В том случае, если система оценка качества является надстройкой над генератором, проблем с определением периода нет – расстояние в тактах между двумя повторяющимися внутренними состояниями генератора и будет период. Несколько сложнее определить период, анализируя непосредственно последовательность. В качестве одного из вариантов определения периода в данном случае можно предложить следующий. Запоминается первый элемент последовательности. Двигаясь по последовательности, находим элемент, равный первому. Анализируем следующий за ним, сравнивая со вторым и так далее до конца последовательности. Если все соответствующие элементы совпадают, это означает наличие периодичности. В данном случае необходимо определить значение периода и при тестировании анализировать только элементы, входящие в период, предварительно выдав информационное сообщение. Данный подход может не сработать в случае наличия предпериода или вырожденных циклов, поэтому имеет смысл сделать несколько испытаний, например, пройти от конца последовательности к началу или начать анализ с заданного элемента последовательности. Для того, чтобы сократить время тестирования, рекомендуется включить оценку периода в один из обязательных последовательных тестов, например, в «Проверку равномерности». Данный тест последовательно обрабатывает элементы последовательности, соответственно, может заодно осуществлять определение периодичности;
- определение области тестирования. В ряде случаев имеется необходимость тестировать не всю последовательность, а ее часть (например, предпериод или период), или же отдельные части элементов последовательности (например, младшие биты каждого элемента). Для реализации данной возможности в составе системы должен быть модуль, позволяющий определять начальную и конечную точку тестирования, значения интервалов в том случае,

если элементы последовательности анализируются не последовательно, а через определенные расстояния и так далее. Данный модуль должен быть связан со средствами оценки периода, чтобы иметь возможность задавать в качестве области тестирования период или предпериод.

- настройки параметров тестов. Каждый тест анализирует число появлений/отсутствия определенного набора, который, в свою очередь, может состоять из поднаборов. Кроме того, тест для получения статистики или статистик, может оперировать несколькими такими наборами. В системе должно быть предусмотрено средство, максимально гибко позволяющее настраивать конфигурацию теста. Например, для теста «проверка серий» данная конфигурация может содержать следующие поля:
  - разрядность серии: 1, 2, …, разрядность элемента последовательности;
  - серии смежные/несмежные
  - последовательность анализируется целиком и/или разбивается на блоки и тест применяется к блоку, в последнем случае необходимо также определить параметры блока;
  - уровень значимости теста.
  - расширения для проверки категорий;
  - расширения для проверки отсутствующих наборов;
- настройки параметров тестирования. Данный блок определяет всю политику тестирования. Указывается, в каком порядке применяются тесты, для каких областей, уровень значимости один для всех тестов или для каждого индивидуальный и так далее. Максимальная проработка блока настроек параметров тестов и параметров тестирования обеспечит наиболее эффективное исследование генератора;
- наличие тестов с проверкой категорий. Большинство тестов возвращают интегральную оценку результатов исследований, при этом зачастую сглаживаются локальные статистические отклонения. Для выявления этих отклонение необходимо оценивать не только конечную статистику теста, но и промежуточные статистики. Большинство существующих систем скрывают данные значения, чтобы не перегружать пользователя излишней информацией, однако данная статистика участвует в определении интегральной оценки, а, следовательно, вычисляется, так что получение данных оценок никак не скажется на производительности тестов;
- возможность получения интегральной оценки. Данное требование решает обратную задачу. В ходе исследований может анализироваться сотни и тысячи последовательностей, причем каждая последовательность набором из нескольких десятков тестов. Для ускорения процесса анализа желательно ввести интегральную оценку, которая оценит как прохождение последовательностью набора тестов, так и прохождение испытания генераторов. Каждый тест возвращает значение P-value, равномерно распределенное на интервале [0, 1], следовательно, можно применить критерий Колмогорова-Смирнова или Андерсона-Дарлинга и получить значение P-value для оценки прохождения последовательностью группы тестов. Данные значения, полученные для всех последовательностей тестируемого генератора, опять обрабатываются при помощи указанных критериев и вычисляется значение P-value, характеризующее успешность прохожднения генератором испытания;
- возможность тестирования нескольких последовательностей. Для принятия решения о свойствах генератора недостаточно протестировать одну последовательность – требуется испытание множества последовательностей, как правило, обратно пропорционального уровню значимости теста. В связи с этим, система оценки качества должна иметь возможность тестировать группу последовательностей (в идеале – несколько генераторов с группами последовательностей). Результат должен представлять как отчеты по каждому испытанию, так и интегральную оценку для сокращения времени анализа;
- развернутый отчет по каждому тесту. Учитывая, что параметры тестирования допускают использование различных настроек для различных тестов, необходимо иметь развернутый отчет по каждому тесту. В состав такого отчета должны входить следующие позиции:
  - детальное описание теста;
  - параметры тестирования;
  - область тестирования;
  - значения промежуточных статистик;
  - интегральная статистика;

    - заключение системы о результате прохождения данной последовательностью данного теста;
- общий отчет. Если детальный отчет по каждому тесту применяется для анализа прохождения последовательностью конкретного теста, то общий отчет необходим для просмотра сведений о тестировании генератора/генераторов. В состав общего отчета должны входить следующие позиции:
    - время начала и окончания тестирования;
    - наименование объектов тестирования (то есть наименование генераторов с указанием параметров, использовавшихся при формировании набора последовательностей);
    - параметры тестирования;
    - интегральная статистика прохождения набором последовательностей набора тестов;
    - интегральная статистика прохождения каждой последовательностью набора каждого теста;
    - заключение системы о результатах испытания.
    - В связи с большим количеством информации, представленной в общем отчете, последний желательно сделать многоуровневым. В качестве примера можно предложить следующую градацию по уровням:
    - обобщенный отчет о тестировании группы генераторов;
    - обобщенный отчет о тестировании конкретного генератора;
    - обобщенный отчет о прохождении конкретным генератором конкретного теста;
    - и так далее. Идеальным вариантом общего отчета является ситуация, когда верхним уровнем является отчет о тестировании целиком, а нижним – развернутый отчет по каждому тесту;
- база данных тестирования. Большое количество информации, полученной в результате выполнения тестирования, делает необходимым создание специализированной базы данных. При помощи данной базы будет удобно создавать многоуровневые вложенные общие отчеты, а также осуществлять сравнительный анализ результатов тестирования разных генераторов или разных версий генераторов. Крайне желательно, чтобы данная база была накопительной, то есть постоянно пополнялась результатами новых испытаний. Еще одним неплохим решением является создания сетевой базы результатов тестирования с целью обмена достижениями географически распределенных разработчиков;
- наличие средств оценки корреляции. Система оценки качества может применяться не только к последовательностям, сформированным генераторами псевдослучайных чисел, но и к результатам преобразования нелинейных функций криптосистем. В связи с этим необходимо наличие средств, позволяющих оценить, какие изменения вносит данное криптографическое преобразование на основе анализа зависимости между входом и выходом данного преобразования;
- возможность добавления новых тестов. Каждый генератор обладает оригинальными особенностями, требующими дополнительной проверки при помощи тестов, учитывающих данную особенность. Для того, чтобы каждый раз не переделывать систему должен быть разработан удобный механизм добавления новых тестов, позволяющий оперативно обновлять информацию о настройках, параметрах тестирования и так далее;
- наличие встроенных генераторов псевдослучайных чисел. Оценка свойств генераторов будет происходить значительно эффективнее, если система оценки будет являться надстройкой над генератором или же генератор будет частью системы. В этом случае появится возможность более развернутого применения теоретических критериев, а также оценки свойств периодичности исследуемых генераторов;
- возможность онлайн-тестирования. Обилие всевозможных критериев, как эмпирических, так и теоретических, накапливаемые в ходе тестирования сведения о свойствах тех или иных генераторов – все это делает необходимым создание системы онлайн-тестирования генераторов псевдослучайных чисел. В идеале данная система должна представлять собой клиент-серверную архитектуру. На стороне клиента находится приложение, которое занимается тестированием генераторов псевдослучайных чисел, а также сбором полученной статистики и отправкой последней на сервер. На сервере хранится информация об эмпирических и теоретических критериях, а также результаты тестирования различных генераторов, полученные от пользователей системы. Возможно развитие данной структуры в виде распределенной вычислительной системы, позволяющей обрабатывать большие объемы информации, используя ресурсы всех компьютеров, подключенных к

системе.

Структурную схему программного комплекса для оценки качества стохастических и кроптографических алгоритмов можно представить следующим образом (рис. 1).

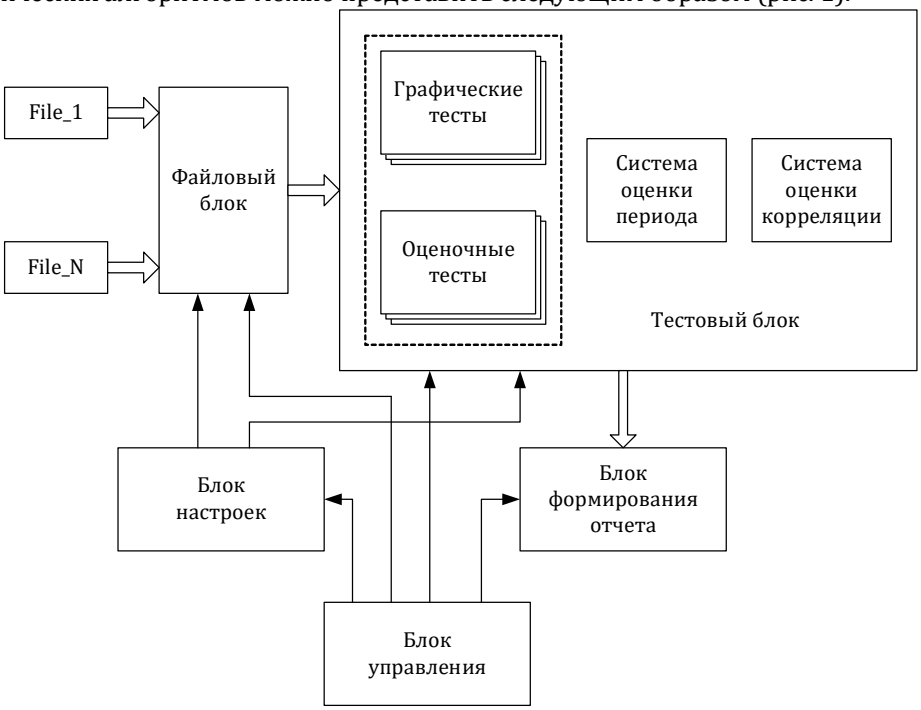

Рис. 1. Структура программного комплекса для оценки качества стохастических и криптографических алгоритмов

В состав комплекса входят следующие блоки:

- файловый блок. Данный блок предназначен для считывания последовательностей из файлов и передачи их для обработки в тестовый блок;
- тестовый блок. Данный блок служит для анализа статистических свойств формируемых ГПСЧ последовательностей. В его состав входят три модуля:
  - система оценки качества – набор графических и оценочных тестов;
  - система оценки периода. Данная программа определяет период анализируемой последовательности. Результаты работы данного модуля используются системой оценки качества для определения длины анализируемого фрагмента последовательности;
  - система оценки корреляции. Данная программа предназначена для определения значения корреляции между файлами;
- блок настроек. Данный блок предназначен для определения имен тестируемых файлов или директорий, а также параметров тестов (границ для P-value, размеров серий и т.д.) и параметров тестирования (размера области тестирования, набора тестов и т.д.);
- блок отчета. Данный блок предназначен для просмотра, записи и печати результатов тестирования;
- блок управления. Данный блок служит для согласованной работы всех блоков устройств и для организации взаимодействия с пользователем.

В настоящее время реализована большая часть данного программного комплекса, а именно: система оценки качества (все тесты Руководства NIST, DIEHARD, Д.Кнута, CRYPT-X, как оригинальные, так и модифицированные, графические тесты, собственные оценочные и графические тесты), система оценки корреляции, система обнаружения периодичности. Примеры работы программного комплекса приведены на рис. 2-3.

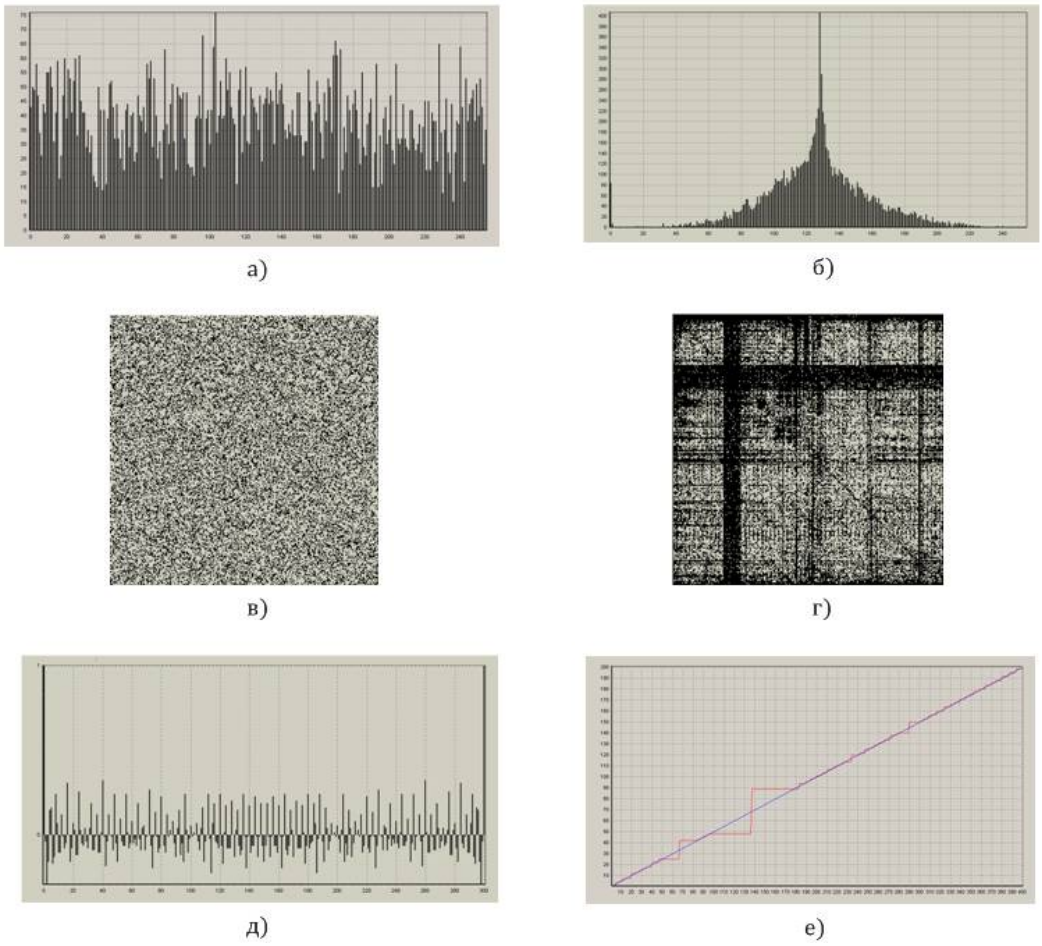

Рис. 2. Примеры графических тестов

Общий отчет

Результаты тестирования файла:
D:\Test\Исследование AES\AES_10k
Размер файла: 10000
Период: 10000
Тестируется: 10000

Тесты	Fэксп	Fтеор	Результат
Проверка 0 и 1	2,33280	6,63500	+
Проверка 0 и 1 в подпослед.	0,44542	0,01000	+
Проверка несцепленных серий			+
Проверка сцепленных серий			-
Проверка дырок	0,41949	0,01000	+
Проверка дырок в подпослед.	0,59132	0,01000	+
Проверка непересек. шаблонов	0,99999	0,01000	+
Проверка пересек. шаблонов	1	0,01000	+
Проверка частот	243,276	310,900	+
Посимвольная проверка			-
Проверка интервалов			+
Проверка перестановок			+
Проверка на монотонность			+
Проверка рангов матриц 32*32	0,17772	0,01000	+

OK

Рис. 3. Пример общего отчета о тестировании генератора

В качестве дальнейших направлений исследований можно предложить следующие:

- уменьшение объема памяти, требуемой для реализации теста. Практически все оценочные тесты производят расчет тестовой статистики на основе частоты встречаемости того или иного вида шаблона. Соответственно, чем длиннее тестируемая последовательность, тем больший объем памяти необходим для хранения числа появлений шаблона определенного типа. Рано или поздно может наступить ситуация, когда размер необходимой для реализации теста памяти превысит предельный размер памяти вычислительной системы, что автоматически приведет к невозможности выполнения теста. В качестве варианта решения данной проблемы можно предложить вычисление промежуточных результатов, либо подсчет числа отсутствующих шаблонов [10]. Второй вариант представляется более предпочтительным, так как позволит избежать зависимости объема необходимой для реализации теста памяти от длины исследуемой последовательности.
- повышение быстродействия статистических тестов. Стремительное развитие средств вычислительной техники, повышение скорости передачи данных, а также появление так называемой легковесной криптографии – все эти факторы приводят к необходимости существенно сокращать время выполнения ряда алгоритмов. Особенно это актуально для стохастических и криптографических алгоритмов, ведь зачастую они применяются для преобразования информации «на лету». В качестве одного из вариантов повышения быстродействия тестов можно предложить выполнение последних при помощи вычислительных ресурсов графических процессоров (технология nVidia CUDA) [11-13]. Если число ядер центрального процессора исчисляется единицами, то количество вычислительных узлов современных видеокарт может достигать нескольких сотен, что позволит многократно увеличить скорость выполнения статистических тестов, а также увеличит гибкость процесса тестирования

Исследование стохастических алгоритмов и методов анализа их качества проводились в рамках реализации Программы повышения конкурентоспособности Национального исследовательского ядерного университета «МИФИ».

Литература

1. Чугунков И.В. Методы и средства оценки качества генераторов псевдослучайных последовательностей, ориентированных на решение задач защиты информации. – М.: НИЯУ МИФИ, 2012. – 236 с.
2. Epishkina A., Kogos K. Quantum random number generator for secure communications // 2016 International Siberian Conference on Control and Communications, SIBCON 2016 – Proceedings. – Moscow, 2016. 10.1109/SIBCON.2016.7491712.
3. Silnov D.S., Tarakanov O.V. Analysis of modern attacks on antiviruses // Journal of Theoretical and Applied Information Technology. - Volume 76, Issue 1, 2015, Pages 59-63.
4. NIST: Advanced Encryption Standard (AES) (FIPS PUB 197). National Institute of Standards and Technology (Nov 2001).
5. A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publications 800-22. Revision 1.a. April, 2010.
6. Marsaglia G. DIEHARD: Battery of tests of randomness [Online]. Available: http://stat.fsu.edu/pub/diehard/.
7. Knuth D. The Art of computer programming. Volume 2: Seminumerical Algorithms. Third Edition. – Addison-Wesley, 1997. – 762 pp.
8. Gustafson H., Dawson E., Nielsen L., Caelli W. A computer package for measuring the strength of encryption algorithms // Computer. & Security, – 1994., vol. 13, issue 8. – pp. 687-697.
9. Chugunkov I.V., Muleys R.B. Pseudorandom numbers generators quality assessment using graphic tests // Proceedings of the 2014 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2014. – St. Petersburg, 2014. – pp. 8-13. DOI: 10.1109/ElConRusNW.2014.6839187.
10. Chugunkov I.V., Prokofiev A.O., Strelchenko P.A. The optimization of statistical tests for pseudorandom number generators // Proceedings of the 2016 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2016. – St. Petersburg, 2016. – pp. 190-193. DOI: 10.1109/EIConRusNW.2016.7448151.
11. Chugunkov I.V., Dyumin A.A., Maksutov A.A., Smirnova I.S. Parallelization of test for assessing pseudorandom number generators using CUDA technology // Proceedings of the 2015 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2015. – St. Petersburg, 2015. – pp. 60-64. DOI: 10.1109/EIConRusNW.2015.7102232.
12. Dyumin A.A., Kuznetsov A.A., Rovnyagin M.M. Evaluation of statistical properties of a modified Bloom filter for heterogeneous GPGPU-systems // Proceedings of the 2015 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2015. – St. Petersburg, 2015. – pp. 190-193. DOI: 10.1109/EIConRusNW.2015.7102234.
13. Vasilyev N.P., Rovnyagin M.M. Software platform VAR for heterogeneous GPGPU-systems // 2016 6th International Conference – Cloud System and Big Data Engineering (Confluence). – Noida, 2016. – pp.624-629. doi: 10.1109/CONFLUENCE.2016.7508194.

References

1. Chugunkov I.V. Metody i sredstva otsenki kachestva generatorov psevdosluchaynykh posledovatel'nostey, orientirovannykh na reshenie zadach zashchity informatsii. – M.: NIYaU MIFI, 2012. – 236 p.

2. Epishkina A., Kogos K. Quantum random number generator for secure communications // 2016 International Siberian Conference on Control and Communications, SIBCON 2016 – Proceedings. – Moscow, 2016. 10.1109/SIBCON.2016.7491712.
3. Silnov D.S., Tarakanov O.V. Analysis of modern attacks on antiviruses // Journal of Theoretical and Applied Information Technology. - Volume 76, Issue 1, 2015, Pages 59-63.
4. NIST: Advanced Encryption Standard (AES) (FIPS PUB 197). National Institute of Standards and Technology (Nov 2001).
5. A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publications 800-22. Revision 1.a. April, 2010.
6. Marsaglia G. DIEHARD: Battery of tests of randomness [Online]. Available: http://stat.fsu.edu/pub/diehard/.
7. Knuth D. The Art of computer programming. Volume 2: Seminumerical Algorithms. Third Edition. – Addison-Wesley, 1997. – 762 pp.
8. Gustafson H., Dawson E., Nielsen L., Caelli W. A computer package for measuring the strength of encryption algorithms // Computer. & Security, – 1994., vol. 13, issue 8. – pp. 687-697.
9. Chugunkov I.V., Muleys R.B. Pseudorandom numbers generators quality assessment using graphic tests // Proceedings of the 2014 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2014. – St. Petersburg, 2014. – pp. 8-13. DOI: 10.1109/ElConRusNW.2014.6839187.
10. Chugunkov I.V., Prokofiev A.O., Strelchenko P.A. The optimization of statistical tests for pseudorandom number generators // Proceedings of the 2016 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2016. – St. Petersburg, 2016. – pp. 190-193. DOI: 10.1109/EIConRusNW.2016.7448151.
11. Chugunkov I.V., Dyumin A.A., Maksutov A.A., Smirnova I.S. Parallelization of test for assessing pseudorandom number generators using CUDA technology // Proceedings of the 2015 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2015. – St. Petersburg, 2015. – pp. 60-64. DOI: 10.1109/EIConRusNW.2015.7102232.
12. Dyumin A.A., Kuznetsov A.A., Rovnyagin M.M. Evaluation of statistical properties of a modified Bloom filter for heterogeneous GPGPU-systems // Proceedings of the 2015 IEEE North West Russia Young Researchers in Electrical and Electronic Engineering Conference, ElConRusNW 2015. – St. Petersburg, 2015. – pp. 190-193. DOI: 10.1109/EIConRusNW.2015.7102234.
13. Vasilyev N.P., Rovnyagin M.M. Software platform VAR for heterogeneous GPGPU-systems // 2016 6th International Conference – Cloud System and Big Data Engineering (Confluence). – Noida, 2016. – pp.624-629. doi: 10.1109/CONFLUENCE.2016.7508194.

Поступила: 5.10.2016

Об авторах:

Чугунков Илья Владимирович, доцент кафедры «Компьютерные системы и технологии» Национального исследовательского ядерного университета «МИФИ», кандидат технических наук, IVChugunkov@mephi.ru;

Прокофьев Антон Олегович, аспирант кафедры «Компьютерные системы и технологии» Национального исследовательского ядерного университета «МИФИ», AOProkofyev@mephi.ru;

Гриднева Елена Андреевна, магистрант кафедры «Компьютерные системы и технологии» Национального исследовательского ядерного университета «МИФИ»., smedyou@gmail.com;

Матрюхина Екатерина Андреевна, бакалавр кафедры «Компьютерные системы и технологии» Национального исследовательского ядерного университета «МИФИ», Ekaterinamtryukhina@gmail.com.