На правах рукописи

Агафьин Сергей Сергеевич

МЕТОДИКА ГЕНЕРАЦИИ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ ПУТЕМ ИЗМЕРЕНИЯ ВРЕМЕНИ ДОСТУПА К ОПЕРАТИВНОЙ ПАМЯТИ

Специальность: 05.13.19 – Методы и системы защиты информации, информационная безопасность

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Автор: 

Москва – 2015

Работа выполнена в Национальном исследовательском ядерном университете «МИФИ» (НИЯУ МИФИ)

Научный руководитель: Кандидат технических наук, доцент кафедры «Криптология и дискретная математика» НИЯУ МИФИ Смирнов Павел Владимирович

Официальные оппоненты: Доктор физико-математических наук, профессор, профессор кафедры «Информационная безопасность» НИУ ВШЭ Бабаш Александр Владимирович

Кандидат технических наук, старший научный сотрудник ФГУП «ЦНИИХМ» Коркин Игорь Юрьевич

Ведущая организация: Московский государственный университет имени М.В. Ломоносова

Защита состоится «07» октября 2015 г. в 16 часов 30 минут на заседании диссертационного совета Д 212.130.08 на базе Национального исследовательского ядерного университета «МИФИ»: 115409, г. Москва, Каширское ш., д.31. Тел. для справок: +7 (499) 324-87-66, +7 (495) 788-56-99.

С диссертацией можно ознакомиться в библиотеке Национального исследовательского ядерного университета «МИФИ» и на сайте: http://ods.mephi.ru.

Просим принять участие в работе совета или прислать отзыв в двух экземплярах, заверенный печатью организации.

Автореферат разослан «__» _____ 2015 г.

Ученый секретарь диссертационного совета Горбатов В.С.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В связи с интенсивным развитием информационных технологий все большую актуальность приобретают проблемы информационной безопасности, от решения которых во многом зависит успешное функционирование организаций и предприятий. Одной из основных задач, решаемой в рамках синтеза систем защиты информации (СЗИ), является программная реализация алгоритмов защиты информации.

Значительное количество алгоритмов защиты подразумевает использование последовательностей, которые являются непредсказуемыми в рамках заданной модели нарушителя. Для их получения используются компоненты систем защиты информации, которые называются генераторами случайных последовательностей.

Данные генераторы используются для решения большого числа задач защиты информации, таких как: обеспечение конфиденциальности информации, передаваемой по компьютерным сетям, путем генерации секретных ключей для алгоритмов преобразования информации; обеспечение подтверждения подлинности документов путем генерации случайных значений для протоколов электронной подписи; защита приложений от атак внедрения кода путем случайного отображения секций исполняемого файла в адресное пространство, и т.д.

Под непредсказуемостью последовательности понимается отсутствие возможности получения рассматриваемым нарушителем какой-либо части последовательности по любому другому отрезку данной последовательности.

Генераторы случайных последовательностей, которые обеспечивают теоретически обоснованную непредсказуемость, независимо от модели нарушителя, называется генераторами истинно случайных последовательностей. Как правило, они основаны на наблюдении за некоторым истинно случайным процессом: радиоактивным распадом, космическим излучением, атмосферным шумом и т.д.

Использование подобных генераторов в системах защиты информации сопряжено с рядом сложностей: необходимы дополнительные аппаратные модули и постоянный аудит их состояния. Если СЗИ функционируют в виртуальных средах, на мобильных платформам или в рамках любой другой нерасширяемой аппаратной конфигурации, обеспечить выполнение данных условий невозможно.

В связи с этим в системах защиты информации могут применяться генераторы случайных последовательностей, которые основаны не на случайных физических процессах, а на некоторых свойствах систем, характеристики которых не являются непредсказуемыми для произвольного нарушителя, однако обеспечивающий достаточный уровень безопасности в рамках выбранных моделей.

Одним из важнейших свойств генераторов случайных последовательностей является их быстродействие – битовая длина случайной последовательности, генерируемой данным компонентом системы защиты информации за 1 секунду.

Производительность генераторов случайных последовательностей, используемых в современных системах защиты информации, составляет около нескольких килобит в секунду, что является недостаточным значением для ряда приложений, требующих значительной производительности (например, для использования в удаленных серверах электронной подписи).

В связи с этим активное распространение в системах защиты информации получили алгоритмы, называемые генераторами псевдослучайных

последовательностей. Последовательности, полученные с помощью генераторов псевдослучайных последовательностей, вычислительно неотличимы от случайных последовательностей, однако получены путем применения детерминированных алгоритмов.

Обоснование стойкости генераторов псевдослучайных последовательностей принято сводить к стойкости алгоритмов, на базе которых они строятся, например, к стойкости алгоритмов преобразования информации регистрового типа, временной сложности решения задачи дискретного логарифмирования в группах точек эллиптической кривой и т.п.

Однако подобные генераторы обладают ключевым недостатком: энтропия всей генерируемой в рамках одного запроса последовательности полностью равна энтропии инициализирующего значения. Это не позволяет их использовать для решения всех описанных задач защиты информации, а значит, актуальной является задача обеспечения высокой скорости генерации случайных последовательностей.

Данной задачей в последние годы занималось большое количество исследователей из разных стран: Schneier B., Shamir A., Muller S., Попов В.О., Смышляев С.В., Tso T. Lacharme P., Jun B., Engel J., LeBlank D.

В России проблемой построения генераторов случайных последовательностей занимаются такие организации как ООО «КРИПТО-ПРО», ООО «Код безопасности», ЗАО «ОКБ САПР», АО фирма «Актив».

Основными требованиями, предъявляемыми к генераторам случайных последовательностей, используемым в системах защиты информации, являются высокое быстродействие (не менее 1 Кбит/с), обоснование непредсказуемости случайной последовательности и применимость генератора к различным аппаратным конфигурациям систем защиты информации. Однако на данный момент не существует генераторов, которые бы удовлетворяли данным критериям.

Автором данной работы предложена методика генерации случайных последовательностей путем измерения времени доступа процессора к оперативному запоминающему устройству электронной вычислительной системы.

Объектом исследования являются системы защиты информации, функционирующие в рамках вычислительных систем на базе процессоров с x86-совместимой архитектурой.

Предметом исследования являются статистические свойства последовательностей времени выполнения операций доступа к оперативному запоминающему устройству.

Цель диссертационной работы: повысить скорость генерации случайных последовательностей в системах защиты информации, функционирующих в вычислительных системах на базе архитектуры x86.

Методы исследования: теория временных автоматов, теория обработки сигналов, теория вероятностей и математическая статистика.

Научная задача заключается в разработке метода эвристической амплитудной демодуляции цифрового сигнала, представляющего собой последовательность измерений времени доступа к оперативному запоминающему устройству.

Для достижения поставленной цели необходимо:

− провести анализ существующих методов генерации случайных последовательностей для средств защиты информации;

− выделить требования, предъявляемые к методам генерации случайных последовательностей;

− построить модель нарушителя, от воздействия которого должен быть защищен разрабатываемый генератор;

− разработать методику генерации случайных последовательностей для программных реализаций алгоритмов защиты информации с использованием основных компонентов вычислительной системы;

− провести оценку статистических свойств генерируемых случайных последовательностей.

Научная новизна работы состоит в следующем:

– построена автоматная модель функционирования оперативного запоминающего устройства, основанная на теории временных автоматов, отличающаяся от известных моделей оперативного запоминающего устройства простотой описания процессов последовательного обращения к памяти, которая позволила изучить процесс взаимодействия основных компонентов вычислительной системы;

– предложена методика генерации случайных последовательностей, основанная на измерении времени взаимодействия основных компонентов вычислительной системы, отличающаяся от известных повышенным быстродействием и применимостью для большинства аппаратных конфигураций вычислительных систем на базе архитектуры x86;

– получена оценка статистических свойств случайных последовательностей, генерируемых на основе измерения времени взаимодействия основных компонентов вычислительной системы, демонстрирующая, что генераторы, построенные на базе предложенной методики, полностью соответствуют требованиям, выделяемым для генераторов случайных последовательностей, используемых в системах защиты информации.

Практическая значимость результатов заключается в том, что предлагаемая методика генерации случайных последовательностей может применяться в системах защиты информации, решающих задачи обеспечения конфиденциальности в условиях противодействия нарушителю, соответствующему модели Н3 ФСБ России, для:

– повышения эффективности СЗИ путем повышения скорости процесса генерации случайных последовательностей;

– повышения стойкости СЗИ на различных аппаратных конфигурациях архитектуры x86 в связи с независимостью предлагаемой методики от вида конфигурации.

Достоверность результатов, обеспечивается корректным применением математического аппарата. Теоретические результаты согласуются с экспериментальными данными. Достоверность экспериментальных данных подкрепляется проведением опытов в полном соответствии с методическими рекомендациями. Допущения и ограничения, принятые в доказательствах утверждений, достаточно обоснованы.

Внедрение результатов исследования. Полученная методика генерации случайных последовательностей для систем защиты информации была использована для разработки нового генератора случайных последовательностей в компании ООО «КРИПТО-ПРО».

Полученная методика генерации случайных последовательностей для систем защиты информации была использована для повышения эффективности процесса тестирования средств защиты информации в АО фирма «Актив».

Теоретические результаты исследования, полученные в процессе выполнения диссертационной работы, использованы в курсе «Разработка и эксплуатация защищенных автоматизированных систем» кафедры «Криптология и дискретная математика» НИЯУ МИФИ для создания лабораторных работ и лекционного материала.

Публикации и апробация работы: Результаты диссертации были изложены в 14 опубликованных и приравненных к ним работах, в том числе 6 статьях в журналах, входящих в Перечень рецензируемых научных журналов, рекомендованных ВАК РФ для публикации основных научных результатов диссертаций на соискание ученых степеней, одна в издании, индексируемом международной системой научного цитирования Scopus, а 7 являются свидетельствами о государственной регистрации программ для ЭВМ. Результаты работы докладывались на конференциях и семинарах различного уровня:

– Международная конференция «SinConf», Глазго, Шотландия, Великобритания, 2014 г.;

– Международная конференция «РусКрипто'15», Московская область, Солнечногорский район, 2015 г.;

– Международный форум по практической безопасности «Positive Hack Days V», г. Москва, 2015 г.;

– 18-я Международная телекоммуникационная конференция студентов и молодых ученых «Молодежь и наука», г. Москва, 2015 г.;

– XX Всероссийская научно-практическая конференция «Проблемы информационной безопасности в системе высшей школы», г. Москва, 2014 г.;

– семинар в Московском Государственном Университете им. М.В.Ломоносова, г.Москва, 2015 г.

Основные положения, выносимые на защиту:

− автоматная модель оперативного запоминающего устройства, используемого в вычислительных системах, позволяющая провести анализ процесса доступа центрального процессора к оперативной памяти;

− метод проведения эвристической амплитудной демодуляции цифрового сигнала, позволяющий выделить шумовую составляющую сигнала с заведомо неизвестными характеристиками;

− методика генерации случайных последовательностей путем измерения времени взаимодействия основных компонентов вычислительной системы, позволяющая использовать частоту процессора в качестве источника энтропии для ГСП, используемых в системах защиты информации.

Структура и объем работы. Работа состоит из введения, четырех глав, заключения, списка литературы, включающего 126 наименований, и трех приложений. Текст диссертации изложен на 137 страницах, включая 12 рисунков и 15 таблиц.

СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность темы диссертационной работы, выделяются и формулируются цель и задачи исследования, описывается структурно-логическая схема диссертационной работы.

В **первой главе** проводится исследование методов генерации случайных последовательностей. Описываются подходы к построению генераторов псевдослучайных последовательностей, а также методы тестирования случайных и псевдослучайных последовательностей на неотличимость от истинно случайных последовательностей.

Базовым подходом к получению случайных последовательностей является генерация значений путем наблюдения за характеристиками физического случайного процесса. Для этого, как правило, используется дополнительное оборудование. Однако этот подход не применим в случаях, не подразумевающих использование дополнительных плат расширения.

В настоящее время существует несколько видов генераторов случайных последовательностей, базирующихся на свойствах основного оборудования вычислительных систем и характеристиках операционных систем:

- биологические;
- на основе фиксации аппаратных прерываний;
- на основе измерения времени отклика оборудования;
- на основе измерения погрешности взаимодействия счетчиков времени;
- на основе измерения фазового дрожания сигнала.

Все они успешно проходят наборы статистических тестов, и генерируемые на их основе последовательности демонстрируют практическую вычислительную неотличимость от истинно случайных последовательностей.

В основе биологических генераторов случайных последовательностей лежат особенности взаимодействия пользователя с вычислительной системой с помощью устройств ввода информации. Измеряемые величины могут быть различными: интервалы между нажатиями на клавиши, положение курсора мыши и направление его движения, время между нажатиями на различные области сенсорного экрана и т.д. При корректной реализации и полноценном обосновании подобные генераторы могут использоваться в любой системе защиты, не подразумевающей внутреннего нарушителя. Однако они обладают существенным недостатком: для их работы требуется непрерывное взаимодействие с пользователем, что, как следствие, влечет низкую скорость генерации. Для некоторых вариантов использования это ограничение может быть критическим: серверное оборудование, высокоскоростные протоколы защиты информации и т.д.

В основе методов, использующих фиксацию аппаратных прерываний процессора, лежит предположение, число аппаратных прерываний, произошедших за определенный интервал времени, а также время их появления являются случайными величинами. Подобный подход частично используется в распространенных операционных системах и подразумевает непрерывный набор информации о прерываниях в процессе работы системы. Это приводит к невозможности получения случайных последовательностей в начале работы с системой. Также в процессе исследования не было найдено ни одного теоретического обоснования, позволяющего считать данный подход применимым в системах защиты информации.

В основе метода измерения погрешности взаимодействия двух счетчиков времени лежит принцип измерения показаний счетчика с низким разрешением счетчиком с высоким разрешением. Это приводит к тому, что появляется возможность измерения нестабильности работы счетчиков, а также различных эффектов вычислительных систем, что, в силу их предполагаемой непредсказуемости, приводит к возможности получения случайности. Данный метод обладает недостатком, связанным с большой зависимостью статистических свойств последовательностей от видов и моделей используемых счетчиков.

Методы на основе времени отклика от периферийного оборудования основываются на рассмотрении в качестве случайных значений времени доступа к различному низкоскоростному оборудованию (чаще всего, жесткому диску). Их недостатки заключаются в сложности описания самого процесса доступа, а также в неприменимости в общем виде для разных наборов оборудования.

Наибольший интерес представляют методы на основе измерения фазового дрожания сигнала, однако в данной области не было проведено достаточное количество исследований и не были получены результаты, которые бы позволили судить о применимости данных методов в системах защиты информации.

Результаты сравнения изученных методов представлены в таблице 1.

Таблица 1 – Сравнение методов генерации случайных последовательностей с использованием штатных компонентов вычислительных систем

Анализируемый метод	**Критерий сравнения**		
	Наличие теоретического обоснование непредсказуемости	Производительность (бит/с)	Работоспособность на всех аппаратных конфигурациях
Биологические генераторы	+	~20	–
На основе фиксации прерываний	–	$\sim 10^3$	+
На основе измерения времени отклика периферийного оборудования	–	$\sim 2 * 10^3$	–
На основе измерения погрешности счетчиков времени	–	$\sim 4 * 10^3$	–
На основе измерения фазового дрожания цифрового сигнала	–	$\sim 10^3$	+

Анализ применимости к аппаратным конфигурациям проведен для систем защиты информации, функционирующих на базе архитектуры x86. Она была выбрана в связи со своей широкой распространенностью среди пользовательских и серверных платформ.

По результатам проведенного анализа было выявлено, что все рассмотренные методы генерации случайных последовательностей с помощью основных компонент ЭВМ обладают существенными недостатками. Предлагаемая в работе методика генерации лишена выявленных недостатков.

Во **второй главе** строится модель нарушителя, относительно которой будет проводиться разработка методики. Предлагается модель оперативного запоминающего устройства в терминах теории временных автоматов. Приводятся результаты исследования процессов, протекающих в вычислительных машинах, которые могут рассматриваться как случайные в рамках построенной модели нарушителя. Излагается созданная методика генерации случайных последовательностей путем измерения времени доступа процессора к оперативной памяти.

В диссертационной работе рассматривается функционирование системы защиты информации в рамках противодействия нарушителю Н3 согласно «Методическим рекомендациям по обеспечению с помощью криптосредств безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств автоматизации», утвержденным приказом № 149/54-144 руководства 8 Центра ФСБ России от 21 февраля 2008 года. Модели нарушителя более высокого уровня (начиная с Н4) не рассматривались, так как включают в себя внутреннего нарушителя с правами администратора системы, противодействие которому на программном уровне невозможно.

Согласно данной модели нарушитель располагает доступными в свободной продаже компонентами системы защиты и документацией на нее, а также способен:

– использовать средства защиты информации в пределах контролируемой зоны на уровне пользователя операционной системы;

– подключать дополнительное оборудование к сетям энергоснабжения, связанным со средством защиты информации;

– запускать в процессе работы генератора случайных последовательностей на вычислительной машине дополнительные пользовательские вычислительные процессы;

– проводить базовый анализ статистических характеристик случайных последовательностей.

Для обеспечения безопасности генератора случайных последовательностей в соответствии с приведенной моделью нарушителя в диссертационной работе проводится изучение процессов, протекающих в вычислительной системе, с целью выделения тех, которые могут быть использованы генератором случайных последовательностей.

Среди них можно выделить: время исполнения операций на основных компонентах системы (центральный процессор, оперативное запоминающее устройство, видеоадаптер, постоянное запоминающее устройство), скорость переключения внутренних состояний данных компонентов, временные характеристики, связанные со сторонними процессами (наведенные).

В типовых вычислительных задачах обращение к оперативному запоминающему устройству (ОЗУ) является одной из самых продолжительных операций. Время ее исполнения на несколько порядков превышает длительность исполнения стандартных арифметико-логических операций центральным процессором. В связи с этим, время обработки запроса данных на чипе ОЗУ должно быть минимальным. Это достигается с помощью различных методов стабилизации времени обработки команд.

Для описания работы оперативного запоминающего устройства его можно представить в виде временного автомата

$$M = (L, I^*, E, T, guard, inv, X, Q), \quad (1)$$

где L – множество позиций автомата M; $I^* \in L$ – начальная позиция автомата; E – множество ребер автоматного графа с пометками, отражающими условие перехода; T – часы, синхронизированные с тактовой частотой системной шины; $guard$ – функция, которая сопоставляет каждому ребру часовое ограничение; inv – отображение, определяющее инвариант для каждого состояния; X – множество входных значений; Q – множество значений длин очереди исполнения команд. T_{ras}, T_{rcd}, T_{rp} – величины, являющиеся характеристиками конкретной оперативной памяти, определенные в технической документации.

В связи с тем, что интерес представляют не одиночные измерения, а последовательности, которые можно интерпретировать как случайные, далее модель используется для описания процесса последовательного чтения набора элементарных блоков данных.

Пусть запросы на чтение посылаются процессором через равные промежутки времени (числа тактов оперативной памяти) δ', полностью определяемые фиксированной частотой процессора ν'_{cpu}: $\delta' \sim \frac{1}{\nu'_{cpu}}$.

Любая операция чтения данных из памяти будет начинаться с команды, которая может повлиять на переход, только если оперативное запоминающее устройство находится в позиции I^* при нулевой длине очереди команд. В рамках данной модели получено, что T_m – время, требуемое на ее полное исполнение, будет периодической величиной, период которой равной НОК$(T_{ras} + T_{rp}, \delta')$. Время исполнения текущей инструкции с номером n будет определяться как $T_Q + T_R$, где T_Q – время, которое потребуется для исполнения инструкций, пришедших ранее и находящихся в данный момент в очереди, а T_R – время, которое потребуется для завершения исполнения текущий выполняемой операции. Тогда, обозначив для простоты $T_F = T_{ras} + T_{rp}$, получим, что $T_Q = T_F(n - \left(\frac{n\delta\prime}{T_F}\right))$, а $T_R = T_F - (n\delta' \ (mod\ T_F))$.

Таким образом, полное время, требуемое для выполнения операции обработки запроса оперативным запоминающим устройством при отсутствии переключения на другой чип данных, равно

$$T_m = T_F + n(T_F - \delta') - (n\delta' \ (mod\ T_F)). \quad (2)$$

Данное равенство справедливо для последовательного запроса данных, хранимых на ОЗУ, при $T_F \geq \delta'$. Если $T_F < \delta'$, время исполнения каждой инструкции

будет равно T_F. Поскольку в реальных системах T_F превышает δ примерно в 100 раз, полученное соотношение можно считать справедливым для любого последовательного запроса данных.

Отсюда очевидно, что если величины δ' и T_F являются неизменяемыми, то время доступа остается периодической величиной без возможности рассмотрения ее в качестве случайной.

Однако в действительности значение времени между последовательными запросами процессора зависит от большого числа факторов, часть которых не поддается математическому описанию без специальных инженерных исследований в силу своей сложности и непредсказуемости.

Среди них можно выделить следующие базовые, которые дают предпосылки к возможности рассмотрения данного значения как случайного.

Во-первых, это несовпадение частот оперативного запоминающего устройства и процессора, приводящее к тому, что при инициировании процессором команды на чтение памяти при константном времени отправки этого сообщения в контроллер, запрос будет передан в само оперативное запоминающее устройство только при начале следующей высокой составляющей меандра.

Во-вторых, наличие промежуточных узлов между процессором и ОЗУ, связанное с наличием наводок от высокочастотного генератора в блоке питания, а также погрешностей в проводимости каналов системной шины вносят существенную погрешность в значение временного интервала между двумя последовательными запросами на чтение.

В-третьих, отсутствие блока, отвечающего за стабилизацию частоты процессора. Процессоры имеют аппаратные ограничители, срабатывающие на резкое повышение напряжения или температуры и удерживающие частоту в плавающем диапазоне (величина данного диапазона составляет 3% от максимальной частоты).

В связи с этим, можно заключить, что частота процессора в некоторый момент времени t (номер такта оперативной памяти) можно описывать случайным блужданием $\nu_{cpu}(t) = \nu'_{cpu} + X(t)$, где $X(t)$ – случайная величина. Так как величина временного интервала между последовательными запросами на чтение из оперативной памяти существенно зависит от частоты процессора ($\delta(t) \sim \frac{1}{\nu_{cpu}(t)}$), то она также является случайной.

Таким образом, можно судить, что величина $\delta(t)$ зависит от взаимодействия ряда периодических процессов и нескольких случайных (прежде всего, нестабильности частоты). Следовательно, ее измерение и дальнейшее исключение периодических составляющих позволит получить случайные значения, генерируемые с высокой скоростью (сравнимой со скоростью доступа процессора к ОЗУ).

В связи с тем, что возможность прямого измерения величины интервалов между операциями доступа к памяти, отсутствует в силу архитектурных особенностей современных вычислительных машин, следует найти способ косвенного измерения данной величины.

Из уравнения (2) очевидно, что величиной, которая наиболее простым образом зависит от δ, является T_m. Поскольку все входящие в нее элементы являются константными либо предсказуемыми, ее также можно рассматривать как случайную, статистические свойства которой определяются через свойства δ.

В связи с тем, что современные вычислительные машины функционируют, как правило, в защищенном режиме, с пользовательского уровня отсутствует возможность точного измерения времени. Также при штатной работе ЭВМ включены технологии аппаратной виртуализации и многопоточности, которые могут вносить непредсказуемую погрешность в измерения.

В связи с этим для проведения измерений необходимо выполнить ряд предварительных шагов, связанных с приведением функционирования системы к предельно простому виду: отключение аппаратной виртуализации, поддержки многоядерной работы и режимов эмуляции многоядерности (Hyper-Threading), всех технологий энергосбережения. В ходе диссертационной работы было установлено, что включение данных технологии может улучшить статистические свойства последовательностей, однако это свойство требует дополнительного изучения.

После отключения всех механизмов процессора, которые могут повлиять на измерение времени доступа процессора к оперативному запоминающему устройству, можно провести его измерение с помощью следующего алгоритма:

1) Отключение обработки аппаратных прерываний и кэширования процессора.
2) Выполнение сериализующей операции для очистки конвейера процессора.
3) Измерений текущего значения счетчика тактов процессора.
4) Сохранение полученного значения счетчика тактов во временный регистр.
5) Измерение нового значения счетчика тактов процессора.
6) Выполнение барьерной и сериализующей операций для защиты конвейера от выполнения инструкций, не входящих в исследуемый блок.
7) Вычисление разницы между значениями, полученными на шагах 5 и 3.
8) Включение кэширования процессора и обработки аппаратных прерываний.

Для набора достаточного числа значений шаги 2-7 следует выполнить большое количество раз. Поскольку операции выполняются с отключенной кэш-памятью процессора, результат измерений будет содержать время доступа к сегменту инструкций процессора.

Полученную последовательность измерений времени доступа к памяти можно считать цифровым сигналом. Она несет в себе как случайную составляющую, так и периодическую, внесенную рядом процессов, описанных ранее. То есть, данный сигнал получен в результате амплитудной модуляции совокупности низкоамплитудных сигналов, имеющих случайную природу, высокоамплитудными периодическими составляющими.

Точная демодуляция методами теории цифровой обработки сигналов в рамках данной работы невозможна в связи со сложностью описания совокупности полезных периодических сигналов. В связи с этим, было принято решение о разработке эвристического метода демодуляции. Он должен отвечать следующим требованиям: во-первых, пороговые значения для критерия исключения периодических составляющих должны выбираться независимо для каждой полученной последовательности; во-вторых, после фильтрации должна сохраняться большая часть случайной составляющей сигнала.

Наиболее изученным и широко распространенным методом разложения цифрового сигнала на гармоники по амплитуде является дискретное преобразование Фурье. Полученный спектр будет описывать периодические составляющие, входящие в сигнал, а амплитуда каждой из полученных гармоник будет описывать степень

влияния сигнала с соответствующей гармонике частотой на результирующую совокупность.

В связи с этим после получения спектра Фурье предлагается провести его фильтрацию с целью исключения периодических составляющих с амплитудой, превышающей амплитуды шумовых составляющих. Вид фильтрации может быть произвольным, однако в целях упрощения процесса предлагается использование следующего подхода: если амплитуда некоторой гармоники превышает заданный порог, то амплитуда среднему значению амплитуд, не превышающих рассматриваемый порог.

Для определения данного порога предлагается следующий метод.

Пусть $N -$ число измерений времени доступа к оперативной памяти; $\{x_k\} -$ множество этих измерений; $\{X_n\} = \text{ДПФ}(x_k) -$ результат дискретного преобразования Фурье.

Зададим индикатор $I_k^p = \begin{cases} 1, \text{ если } |X_k| \geq p, \\ 0, \text{ если } |X_k| < p. \end{cases}$, который принимает значение 1, если амплитуда заданной гармоники превышает некоторое пороговое значение $p \in R^+$. Тогда число гармоник, амплитуда которых превышает пороговое значение p, определяется как

$$C_p = \sum_k I_k^p. \tag{3}$$

Доля гармоник θ_p, амплитуда которых попадает в интервал $[p, 1.05 * p]$, рассчитывается как

$$\theta_p = \frac{C_p - C_{1.05p}}{N}. \tag{4}$$

При $p = 0$ это значение равно нулю. В связи с тем, что гармоники с крайне низкой амплитудой не могут быть учтены, их число в полном спектре будет незначительным. Следовательно, при малых p значение θ_p также будет малым.

При значениях p, приближающихся к максимальному значению амплитуды, и равных ему, значение θ_p будет стремиться к 1 в связи с малым числом периодических составляющих.

Пусть f – функция, определяющая зависимость между θ_p и p: $\theta_p = f(p)$. Эта функция сначала монотонно растет, а затем монотонно убывает. Следовательно, существует значение p', являющееся наибольшим значением функции (глобальным экстремумом).

В связи с тем, что число шумовых составляющих значительно превышает число периодических составляющих, а также их амплитуды имеют близкие значения, уменьшение значения амплитуд, попадающих в заданный интервал, после некоторого значения свидетельствует о том, что это значение является границей амплитуд случайных составляющих.

Зададим множество $X' = \{X_i || X_i| < p'\}$, состоящее из всех гармоник, амплитуда которых ниже рассматриваемого порогового значения.

Пусть $\overline{X'}$ – среднее значение амплитуды среди всех гармоник, входящих в множество X'.

Тогда удаление периодической составляющей будет заключаться в фильтрации гармоник, амплитуда которых превышает значение p', то есть в создании спектра $\{Y'_n\}$:

$$Y'_i = \begin{cases} \overline{X'}, \text{если} |X_i| \geq p', \\ X_i, \text{если } |X_i| < p'. \end{cases} \forall i \in \{0, \dots, N-1\}. \quad (5)$$

Из данного спектра путем применения обратного дискретного преобразования Фурье можно синтезировать дискретный сигнал, который будет содержать случайные шумовые составляющие и низкоамплитудные детерминированные составляющие, не удаленные предложенным методом.

Исходные значения последовательности измерений времени доступа к оперативной памяти, представленные в виде цифрового сигнала, определяются большим количеством факторов. Некоторые из них приводят к тому, что в цифровом сигнале присутствуют высокоамплитудные периодические составляющие, из-за которых может нарушиться базовое требование непредсказуемости случайной последовательности.

После применения предложенного метода амплитудной демодуляции все компоненты сигнала будет незначительно отличаться от шумовых случайных составляющих, что позволяет судить о том, что полученная после применения обратного преобразования Фурье последовательность обладает свойством непредсказуемости в силу вычислительной сложности процесса декомпозиции рассматриваемого сигнала на составляющие компоненты лишь по части последовательности без изучения архитектурных особенностей вычислительной системы.

Предлагается методика генерации случайных последовательностей, состоящая из двух этапов: предварительного и оперативного этапов, основное содержание которых представлено в таблице 2.

В связи с тем, что корректное функционирование всех компонент вычислительной системы, которые используются в предлагаемой методике, является необходимым условием для работы системы защиты информации в штатных режимах, внешние условия, при которых данная методика будет применима, совпадают с условиями, в которых допускается эксплуатация основного оборудования системы согласно документации.

Так как необходимым оборудованием являются только процессор и оперативная память, можно заключить, что применимость данной методики распространяется на все x86-совместимые вычислительные системы, в которых отключена стабилизация частоты процессора при зафиксированном множителе.

Таблица 2 – Основное содержание пошаговой методики генерации случайных последовательностей путем измерения времени доступа к оперативной памяти

Название этапа	**Содержание шагов**
Предварительный	1. Установить и настроить модуль ядра (драйвер) операционной системы, который будет осуществлять генерацию случайных последовательностей. 2. Отключить технологии энергосбережения, многоядерности и аппаратной виртуализации.
Оперативный, на стадии эксплуатации СЗИ	3. Провести набор значений последовательного измерения времени доступа к оперативному запоминающему устройству с помощью предложенного алгоритма. 4. Представив полученные данные в виде дискретного сигнала, применить к ним дискретное преобразование Фурье. 5. Провести фильтрацию спектра Фурье с помощью предложенного метода. 6. Провести обратное преобразование Фурье для получения истинно случайной последовательности.

В **третьей главе** приведено описание архитектуры генератора случайных последовательностей, реализующего предложенную методику генерации.

Для тестирования предложенной методики генерации случайных последовательностей была разработана архитектура программного средства, которое было реализовано для операционных систем семейства Linux. Данная архитектура представлена на рисунке 1.

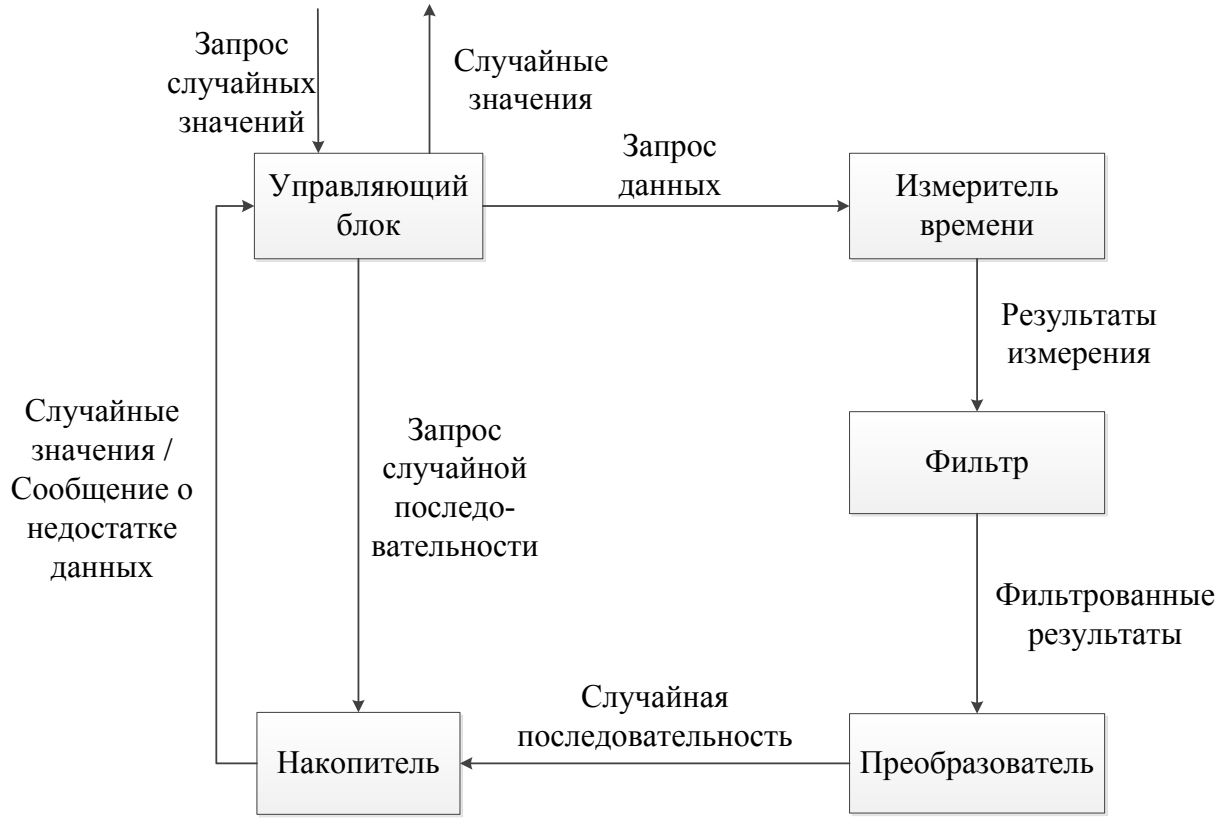

Рисунок 1 – Архитектура генератора случайных последовательностей

Основным компонентом генератора является управляющий блок. Он осуществляет обработку пользовательских запросов и возвращает случайную последовательность, полученную с помощью методики, описанной в данной диссертационной работе. В большинстве случаев предполагается, что запрос приходит от генератора псевдослучайных последовательностей.

Поскольку методика включает в себя отключение обработки аппаратных прерываний на время набора достаточных для дальнейшего анализа данных, постоянное использование ее для генерации последовательностей малой длины является некорректным шагом. Таким образом, предлагается использовать блок накопителя, в котором будут храниться истинно случайные последовательности, которые не были запрошены, но уже успешно сгенерированы в силу блоковой природы генератора.

Управляющий блок запрашивает последовательность требуемой длины у накопителя и, если он пуст, обращается с запросом на начало генерации к измерителю времени. Этот блок реализует алгоритм набора последовательности измерений времени исполнения инструкций чтения из оперативного запоминающего устройства.

Полученная последовательность передается в преобразователь, выполняющий дискретное преобразование Фурье и удаление из полученного спектра периодических составляющих с большой амплитудой.

Результат обратного дискретного преобразования Фурье для полученного модифицированного спектра запоминается в накопителе для дальнейшей передачи управляющему блоку.

В рамках рассмотренной ранее модели нарушителя было указано, что нарушитель имеет возможности эксплуатации средства защиты информации и вычислительной системы, в рамках которой оно функционирует, на уровне пользователя.

Также нарушитель имеет возможность исследования статистических свойств генерируемой последовательности.

Предлагаемая архитектура подразумевает, что генерация случайной последовательности и хранение полученных значений происходят на уровне ядра операционной системы, что исключает возможность получения дополнительной информации о последовательности.

Навязывание данных исключается в связи с особенностями методики (использование запрета аппаратных прерываний).

Единственным недостатком подхода в рамках модели нарушителя, обладающего доступом к пользовательскому интерфейсу средства защиты информации, является то, что в процессе функционирования генератора система фактически переключается в однозадачный режим, что значительно влияет на рабочие процессы, протекающие как в ЭВМ, так и в самой системе защиты информации. То есть, использование данного ГСП может блокировать остальную штатную функциональность СЗИ, используемых в данной системе, и нарушать целостность информации.

Для противодействия данному воздействию в управляющий блок генератора случайных последовательностей можно включить модуль аудита. Он будет собирать следующую информацию:

– идентификатор запрашивающего процесса;
– подробная информация о вложенности вызовов (стек);
– длина запрошенной последовательности;
– время, потраченное на генерацию требуемой последовательности.

Для превентивного исключения процессов, представляющих потенциальную угрозу доступности данных, может быть реализован модуль принятия решения об отказе заданному процессу в генерации последовательности. Он может функционировать как по принципу «белого списка» (указание администратором перечня легальных приложений, которые могут генерировать последовательности неограниченной длины), так и с помощью эвристических механизмов, заключающихся в анализе активности процесса, в частности времени генерации.

В **четвертой главе** описаны экспериментальные результаты применения методики генерации случайных последовательностей. Приводятся результаты применения к получаемым последовательностям наборов статистических тестов NIST SP 800-22. Приводятся результаты измерения быстродействия предлагаемого генератора на различных тестовых стендах. Изложены результаты внедрения разработанного генератора случайных последовательностей в реальные системы защиты информации.

В ходе диссертационной работы было реализовано программное средство, реализующее предложенную архитектуру, для операционной системы на базе ядра Linux.

Для полученных на тестовых стендах последовательностей с помощью критерия согласия χ^2 с уровнем значимости $\alpha = 0.1$ была принята гипотеза H_0 о том, что они обладают нормальным распределением со среднеквадратичным отклонением, совпадающим с выборочным среднеквадратичным отклонением последовательностей.

Для экспериментальной проверки случайности генерируемых последовательностей с помощью набора статистических тестов NIST SP 800-22 было проведено исследование 10 последовательностей длины 204800 бит. Параметры и результаты тестирования приведены в таблице 3. Для проведения тестирования полученные последовательности были предварительно приведены к равномерному распределению преобразованием, сохраняющим статистические свойства исходных последовательностей.

Тесты, отмеченные «*», имеют несколько вариантов, в связи с чем, в таблице приведены значения с минимальным значением *P-value.*

Как видно из приведенных результатов, полученная последовательность успешно прошла тесты. Отсутствие успеха для некоторых последовательностей объясняется случайной природой последовательности, которая не может полностью совпадать с теоретическими описаниями.

Также был проведен анализ автокорреляционной функции исходных последовательностей времени доступа к оперативной памяти и полученных после фильтрации предложенным методом. Функция автокорреляции исходной последовательности приближается к нулевому значению при задержках, сравнимых с $\frac{N}{1000}$, где N – длина рассматриваемой последовательности.

Таблица 3 – Результаты тестирования NIST SP 800-22

Название и параметры теста	Значение *P-value*	Число последовательностей, прошедших тест
Частотный побитовый тест	0.534	10/10
Частотный блочный тест (m=20480)	0.122	10/10
Тест на последовательность одинаковых бит*	0.122	9/10
Тест на самую длинную последовательность единиц в блоке	0.740	10/10
Тест рангов битовых матриц	0.740	10/10
Спектральный тест	0.213	10/10
Тест на совпадение неперекрывающихся шаблонов* ($m = 9$)	0.067	9/10
Тест на совпадение перекрывающихся шаблонов ($m = 9$)	0.350	10/10
Универсальный статистический тест Мауэра	0.534	10/10
Тест на линейную сложность ($M = 1000$)	0.534	10/10
Тест на периодичность ($m = 16$)	0.350	10/10
Тест приблизительной энтропии ($m = 10$)	0.213	10/10
Тест кумулятивных сумм*	0.122	9/10
Тест на произвольные отклонения*	0.036	10/10
Другой тест на произвольные отклонения*	0.074	10/10

Более 95% значений функции автокорреляции последовательностей, полученных после фильтрации, попадают в диапазон $[-\frac{1}{N} - \frac{2}{\sqrt{N}} : -\frac{1}{N} + \frac{2}{\sqrt{N}}]$, где N – длина последовательностей. Это означает, что автокорреляционные свойства генерируемых предложенной методикой последовательностей близки к автокорреляционным свойствам истинно случайных последовательностей.

Данный результат можно считать независимым подтверждением того, что предлагаемая методика позволяет генерировать последовательности, элементы которых являются взаимно независимыми и, следовательно, полученную последовательность можно считать непредсказуемой влево и вправо.

Так как полученные последовательности обладают нормальным распределением, энтропия h байтовых элементов этих последовательностей определяется значением среднеквадратичного отклонения σ:

$$h = \ln(\sigma\sqrt{2\pi e}). \quad (6)$$

Таким образом, если T – полное время (в секундах), затраченное на генерацию одной случайной последовательности, а N – суммарная битовая длина этой последовательности, то производительность v предложенного генератора можно рассчитать как:

$$v = \frac{Nh}{T} = \frac{N * \ln\left(\sigma\sqrt{2\pi e}\right)}{T}. \quad (7)$$

Производительность генератора полностью определяется моделями процессора, оперативного запоминающего устройства и частотами, на которых они функционируют. Для каждого экспериментального стенда были рассчитаны значения производительности и занесены в таблицу 4.

Таблица 4 – Результаты измерения производительности ГСЧ

Описание стенда	Производительность
ЦП: AMD Athlon FX-4200 Zambezi (Bulldozer), 3300 МГц ОЗУ: Kingston HyperX, DDR3, 1866 МГц, 8 Гб, 1 плата	43,13 Кбит/с
ЦП: Intel Core i7-2620M (Sandy Bridge), 2700 МГц ОЗУ: Elpida DDR3, 1866 МГц, 8 Гб, 1 плата	38,96 Кбит/с
ЦП: Intel Core i5-4590 (Haswell), 3300 МГц ОЗУ: Corsair DDR3, 2400 МГц, 16 Гб, 2 платы	41,39 Кбит/с

Полученные значения в среднем в 10 раз превышают производительность существующих генераторов случайных последовательностей и позволяют судить о том, что цель повышения быстродействия генерации случайных последовательностей была достигнута.

Были проведены исследования влияния на статистические характеристики последовательности, генерируемой разработанным ГСП, изменения напряжения процессора в эксплуатационных диапазонах, а также изменения внешней нагрузки в сетях электроснабжения. Наличие стабилизаторов блоков питания, соответствующих стандарту ATX 2.0, а также отсутствие зависимости частоты процессора от напряжения процессора, привели к полному отсутствую влияния данных изменений на статистические характеристики.

Требование запрета обработки аппаратных прерываний также позволяет ограничить воздействие на генератор со стороны пользовательских процессов: планировщик операционной системы, использующий для переключения контекстов процессов прерывания, не сможет передать управление другому процессу.

Тем самым, можно заключить, что полученный генератор полностью соответствует выбранной модели нарушителя и может быть использован в средствах защиты информации, противодействующих ей.

Полученная в диссертационной работе программная реализация генератора случайны последовательностей была использована при разработке системных библиотек «КриптоПро CSP 4.0» в ООО «КРИПТО-ПРО», в системе модульного тестирования АО фирма «Актив», а результаты исследования существующих методов генерации случайных последовательностей были внедрены в курс «Разработка и эксплуатация защищенных автоматизированных систем» на кафедре «Криптология и дискретная математика» НИЯУ МИФИ.

В **заключении** приведены основные результаты диссертационной работы, представлены выводы, полученные в ходе выполнения работы, а так же рассмотрены пути дальнейшего развития темы исследования.

ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ РАБОТЫ

1. Проведен анализ существующих методов генерации случайных последовательностей. Обоснована необходимость разработки нового подхода к генерации случайных последовательностей, применимого для ЭВМ произвольной конфигурации.
2. Построена модель нарушителя, описывающая возможности и инструменты, с помощью которых может осуществляться противодействие генераторам случайных последовательностей. Данная модель позволила выделить основные требования, предъявляемые к разрабатываемому генератору случайных последовательностей.
3. Предложена в терминах теории временных автоматов модель оперативного запоминающего устройства, описывающая процесс обработки запроса данных. Предложенная модель позволила выявить закономерности длительностей выполнения запроса данных, что позволило построить методику генерации случайных последовательностей.
4. Предложен универсальный алгоритм измерения времени доступа к оперативному запоминающему устройству. Данный алгоритм не зависит от конкретной аппаратной конфигурации вычислительной системы и позволяет получать последовательность, содержащую случайную составляющую.
5. Предложен основанный на теории цифровой обработки сигналов метод исключения периодической составляющей из последовательности обращений к оперативной памяти, который позволил выделить случайную составляющую данного процесса.
6. Разработана методика генерации случайных последовательностей на основе измерения времени доступа к памяти, состоящая из двух этапов: предварительного и оперативного. На предварительном этапе проводится установка и настройка модулей генератора, а также оценка применимости методики к данной ЭВМ. На оперативном этапе осуществляется генерация случайных последовательностей по запросу от пользовательских приложений.
7. Предложена архитектура и разработано программное средство генерации случайных последовательностей для некоторых операционных систем, и проведено тестирование предложенной методики на нескольких испытательных стендах. Данное тестирование позволило подтвердить случайный характер последовательностей, генерируемых предлагаемым генератором.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Agafin S. S., Krasnopevtsev A.A. Memory access time as entropy source for RNG // SIN '14 Proceedings of the 7th International Conference on Security of Information and Networks. (Публикация входит в систему цитирования Scopus);
2. Агафьин С.С. Обеспечение доверенности клиентской стороны в системах интернет-банкинга путем использования отчуждаемых носителей ключевой информации // Безопасность информационных технологий. №2. 2013. стр. 8-9. (Журнал входит в перечень ВАК);
3. Агафьин С.С. Верификация программного обеспечения отчуждаемых носителей ключевой информации // Материалы XX Всероссийской научно-практической конференции «Проблемы информационной безопасности информационной безопасности в системе высшей школы». Безопасность информационных технологий. №1. 2013. стр. 74. (Журнал входит в перечень ВАК);

4. Агафьин С.С., Краснопевцев А.А., Смирнов П.В.. Обнаружение недокументированных возможностей приложений // Безопасность информационных технологий. № 3. 2013. стр. 72-74. (Журнал входит в перечень ВАК);

5. Агафьин С.С., Краснопевцев А.А. Атаки на программное обеспечение внешних аппаратных ключевых носителей // Безопасность информационных технологий. №1. 2014. стр. 5-8. (Журнал входит в перечень ВАК);

6. Агафьин С.С., Смирнов П.В. Методы обнаружения недекларированных возможностей в программном обеспечении внешних аппаратных модулей // Безопасность информационных технологий. №2. 2014. стр. 5-10. (Журнал входит в перечень ВАК);

7. Агафьин С.С. Функции безопасности внешних аппаратных модулей, используемых для хранения криптографических ключей // Безопасность информационных технологий. №3. 2014. стр. 37-43. (Журнал входит в перечень ВАК);

8. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.6.1)» / Агафьин С.С., Беляев А.А., Беляев А.А., Бородин Г.О., Годин А.А., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Непомнящий П.В., Попов В.О., Русев А.А., Смирнов П.В. – №2013612780, 2013г.

9. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.6.1) вариант исполнения 8» / Агафьин С.С., Беляев А.А., Бородин Г.О., Беляев А.А., Годин А.А., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Непомнящий П.В., Попов В.О., Русев А.А., Смирнов П.В. – №2013613271., 2013г.

10. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.6.1) вариант исполнения 9» / Агафьин С.С., Беляев А.А., Бородин Г.О., Беляев А.А., Годин А.А., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Непомнящий П.В., Попов В.О., Русев А.А., Смирнов П.В. – №2013613271., 2013г.

11. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.6.1) вариант исполнения 10» / Агафьин С.С., Бородин Г.О., Беляев А.А., Годин А.А., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Непомнящий П.В., Попов В.О., Русев А.А., Смирнов П.В. – №2013612779., 2013г.

12. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.8)» / Агафьин С.С., Беляев А.А., Бородин Г.О., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Попов В.О., Смышляев С.В., Русев А.А., Пичулин Д.Н. – №2014610014., 2014г.

13. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 3.9)» / Агафьин С.С., Беляев А.А., Бородин Г.О., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Попов В.О., Смышляев С.В., Русев А.А. – №2014610725., 2014г.

14. Свидетельство о государственной регистрации программы для ЭВМ «Средство криптографической защиты информации «КриптоПро CSP (версия 4.0)» / Агафьин С.С., Беляев А.А., Бородин Г.О., Дьяченко Д.Г., Коллегин М.Д., Леонтьев С.Е., Попов В.О., Смышляев С.В., Русев А.А. – №2014610162., 2014г.

АГАФЬИН СЕРГЕЙ СЕРГЕЕВИЧ

МЕТОДИКА ГЕНЕРАЦИИ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ ПУТЕМ ИЗМЕРЕНИЯ ВРЕМЕНИ ДОСТУПА К ОПЕРАТИВНОЙ ПАМЯТИ

Подписано в печать 16.07.2015. Формат 60 × 84 $^{1}/_{16}$.
Усл. печ. л. 1,0. Уч.-изд. л. 1,0. Тираж 100 экз. Заказ № 1

Национальный исследовательский ядерный университет «МИФИ» (НИЯУ МИФИ)

115409, Москва, Каширское шоссе, 31