ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Зам. директора по работе с НИУ
ООО «1С»

_____ Н.Ю. Старичков
«05» мая 2022г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента
программной инженерии,
канд. техн. наук

_____ В.В. Шилов
« 06 » мая 2022 г.

КРИПТОГРАФИЧЕСКИ СТОЙКИЙ ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.03.10-01 81 01-1-ЛУ

Исполнитель:
студент группы БПИ201
_____ / Камнев П.А. /
«05» мая 2022 г.

Москва 2022

Инв. № подл	RU.17701729.03.10-01 81 01-1-ЛУ
Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
RU.17701729.03.10-01 81 01-1				

КРИПТОГРАФИЧЕСКИ СТОЙКИЙ ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Пояснительная записка

RU.17701729.03.10-01 81 01-1

Листов 17

Москва 2022

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Наименование разработки

Наименование разработки на русском языке: Криптографически стойкий генератор псевдослучайных чисел.

Наименование разработки на английском языке: Cryptographically secure pseudorandom number generator.

1.2. Основание для разработки

Документ, на основании которого ведется разработка:

Основанием для разработки программы (криптографически стойкого генератора псевдослучайных чисел) является Учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия», а также утвержденная академическим руководителем программы тема курсового проекта.

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение

Функциональным назначением разрабатываемой программы является генерация псевдослучайных последовательностей натуральных чисел с возможностью указания количества и диапазона генерируемых чисел, а также генерация псевдослучайных последовательностей бит.

2.2. Эксплуатационное назначение

Программа может быть использована в любых целях, которые требуют генерации равномерно распределенных псевдослучайных последовательностей чисел или псевдослучайных последовательностей бит, в частности, в криптографических целях: для генерации ключей шифрования, для генерации случайных паролей, для генерации токенов доступа. Таким образом, программа позволяет генерировать псевдослучайные последовательности для таких целей, в которой является критической возможность предсказания следующих элементов псевдослучайной последовательности на основе предыдущих.

Также при инициализации «зерном» генератор может быть использован для генерации псевдослучайных данных при тестировании других программ. Инициализированный «зерном» генератор позволяет воспроизводить одну и ту же псевдослучайную последовательность, что может быть полезно при тестировании (когда ошибочный сценарий воспроизводится не при любых наборах входных данных).

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

В процессе выполнения курсового проекта предполагается реализовать консольную утилиту, предоставляющую возможность генерации криптографически стойких псевдослучайных данных:

- генерация псевдослучайных последовательностей бит,
- генерация псевдослучайных чисел.

В рамках поставленной задачи необходимо реализовать алгоритм, позволяющий генерировать криптографически стойкие последовательности псевдослучайных бит, и на основании него возможна генерация псевдослучайных чисел. Также необходимо реализовать интерфейс командной строки, позволяющий пользоваться разработанным генератором.

3.2. Описание алгоритма и функционирования программы

3.2.1 Работа пользователя с программой через интерфейс командной строки

- Взаимодействие пользователя с программой происходит через интерфейс командной строки (консоль).
  Запуск генератора через командную строку происходит следующим образом: *secure-generator.exe [опции] <параметры>;*
- Пользователю доступны следующие опции:
  1) --bits (генерация бит, по умолчанию генерируются числа);
  2) --min (нижняя граница промежутка генерируемых чисел), параметром является целое число в промежутке от 0 до 2^{32} - 1, по умолчанию принимает значение 0;
  3) --max (верхняя граница промежутка генерируемых чисел), параметром является целое число в промежутке от 0 до 2^{32} - 1, по умолчанию принимает значение 2^{32} - 1;
  4) --length (длина генерируемой последовательности, при генерации чисел – количество чисел, при генерации бит – количество байт), параметром является целое число не менее 1, по умолчанию принимает значение 1;
  5) --file (вывод в файл, по умолчанию вывод в консоль), параметром является имя файла со сгенерированными значениями, который будет создан в каталоге, где расположен исполняемый файл;
  6) --seed (инициализировать генератор числом, по умолчанию происходит инициализация энтропией устройства), параметром является целое число в промежутке от 0 до 2^{32} - 1;
  7) --test_mode (режим тестирования), при выборе этой опции генерируются псевдослучайные биты и непрерывно выводятся в консоль до внешнего завершения процесса, единственная доступная опция при наличии данного параметра – инициализация числом.

- В ошибочных сценариях программа выводит пользователю ошибку в консоль и завершает свое выполнение. Возможные ошибки:
  1) В случае ввода некорректных параметров через аргументы командной строки в консоль выводится информация о том, в каком виде задаются параметры, программа завершает свое выполнение. Выводимый текст:
     Доступны следующие опции:
     --bits (генерация бит, по умолчанию генерируются числа)
     --min (нижняя граница промежутка генерируемых чисел), параметром является целое число в промежутке от 0 до 2^32 - 1, по умолчанию принимает значение 0

--max (верхняя граница промежутка генерируемых чисел), параметром является целое число в промежутке от 0 до 2^32 – 1;
--length (длина генерируемой последовательности, при генерации чисел – количество чисел, при генерации бит – количество байт), параметром является целое число не менее 1, по умолчанию принимает значение 1
--file (вывод в файл, по умолчанию вывод в консоль), параметром является имя файла со сгенерированными значениями, который будет создан в каталоге, где расположен исполняемый файл
--seed (инициализировать генератор числом, по умолчанию происходит инициализация энтропией устройства), параметром является целое число в промежутке от 0 до 2^32 - 1
--test_mode (режим тестирования), при выборе этой опции генерируются псевдослучайные биты и непрерывно выводятся в консоль до внешнего завершения процесса, единственная доступная опция при наличии данного параметра – инициализация числом (--seed)

2) При использовании несовместимых опций (--bits и --min или --max, --test_mode и любая другая опция, кроме --seed) выводится ошибка. Шаблон ошибки:
Использованы несовместимые опции <опция 1> и <опция 2>.
3) Если указана опция, но не указан ее параметр (кроме --bits и –test_mode), выводится ошибка по шаблону:
Не указан параметр для опции <опция>.
4) При генерации значений в файл программа проверяет доступное место на диске. Если места на диске недостаточно, в консоль выводится ошибка, программа завершает свое выполнение. Текст ошибки:
На диске недостаточно свободного места для генерации значений в файл. Проверьте параметр --length.
5) При возникновении ошибок, связанных с получением энтропии от устройства, в консоль выводится краткая информация об ошибке, программа завершает свое выполнение. Текст ошибки:
Возникла ошибка при получении энтропии от устройства. Проверьте соответствие системным требованиям.

3.2.2. Алгоритмическая часть

– В основе класса генерации последовательностей псевдослучайных бит лежит криптостойкий генератор псевдослучайных чисел «Hash_DRBG» на основе хеш-функции SHA-256 в соответствии со стандартом NIST SP 800-90A (Специальная публикация Национального института стандартов и технологий (англ. NIST) с названием «Рекомендация для генерации случайных чисел с использованием детерминированных генераторов случайных битов»);

– Если не используется параметр --seed, для инициализации используется источник энтропии, предоставляемый операционной системой (например, в UNIX-подобных системах имеется такой источник, как /dev/random);

– На основе генератора бит реализуется генератор псевдослучайных чисел (применяется private-наследование);

– Для преобразования битовых последовательностей в произвольные числа с равномерным распределением (без «перекоса») необходим специальный подход: Чтобы получить число на промежутке от a до b, где a и b меньше «сырого» генерируемого

числа, достаточно получить число на промежутке от 0 до (b - a) и прибавить к нему a. Очевидным решением задачи получения числа на промежутке [0, r] кажется получение остатка от деления «сырого» числа на (r + 1), но это создает проблему «перекоса» у чисел получается неравномерное распределение – при достаточно больших значениях r разница вероятности появления разных чисел может отличаться в 2 раза, что существенно ухудшает статистические свойства алгоритма.
Высокую эффективность показывает подход с битовым маскированием числа:

1) При помощи битовой маски получаем число в интервале [0, 2^k), где 2^k – наибольшая степень двойки, превосходящая r,
2) Если получившееся число входит в промежуток [0, r], то возвращаем его, иначе генерируем новое и начинаем сначала.

3.3. Организация входных данных

Входными данными являются параметры работы программы, вводимые в виде аргументов командной строки.

3.4. Организация выходных данных

В случае вывода генерируемых последовательностей в файл, создается файл с именем, заданным пользователем через аргументы командной строки. В противном случае генерируемые последовательности выводятся в консоль.

- При генерации последовательностей чисел генерируемые числа выводятся через пробел (без дополнительных выходных данных).
- При генерации последовательностей бит происходит вывод двоичных данных (сгенерированные файлы можно просмотреть при помощи специальных редакторов двоичных данных, например, Hex Editor Neo).

3.5. Описание и обоснование выбора и состава технических и программных средств

3.6.1. Описание технических и программных средств

Рекомендуемые минимальные параметры устройства для стабильного функционирования программы (на основании системных требований к Windows XP SP3):

- Pentium-совместимый процессор с частотой 233 MHz;
- 64 МБ оперативной памяти;
- 1,5 ГБ свободного места на диске;
- клавиатура;
- видеоадаптер и монитор с разрешением Super VGA (800 x 600).

Программа написана на языке C++17.
В исходном коде программы используются:

- стандартная библиотека языка C++;
- криптографическая библиотека OpenSSL с открытым исходным кодом (в ней присутствуют реализации распространенных хеш-функций);
- WinAPI и Microsoft CryptoAPI предоставляют вызовы, позволяющие получить системную энтропию.

3.6.2. Обоснование выбора технических и программных средств

3.6.2.1. Работа пользователя с программой через интерфейс командной строки

Подход с интерфейсом командной строки выбран исходя из области, в которой чаще всего может быть применен псевдослучайный генератор:

- Генерация тестовых наборов данных (в таких случаях, как правило, данные должны быть с генерированы в файл) – нет необходимости в графическом интерфейсе;
- Использование в прикладных решениях – будет удобнее «пробросить» команды в консоль, чем взаимодействовать с графическим интерфейсом через код.

3.6.2.2. Алгоритмическая часть

Для создания пароля необходим криптографически стойкий генератор псевдослучайных чисел (далее ГПСЧ). Анализ стандартов использования ГПСЧ для генерации паролей показал следующее:- В SP 800-90A Rev. 1 предложено несколько алгоритмов генерации псевдослучайных чисел, в их числе HMAC-DRBG, HASH-DRBG, CTR-DRBG (использует 3DES или AES).

- Часто используется стандарт ANSI X9.17 (использует 3DES);
- ISAAC.

Также имеются некоторые российские государственные стандарты, связанные с генерацией случайных чисел:

- ГОСТ Р ИСО 28640-2012 – описываются статистические методы генерации псевдослучайных чисел, которые можно использовать при применении метода Монте-Карло. Не рассматриваются криптографические генераторы;
- ГОСТ Р 28147-89 – описывается блочный алгоритм шифрования, который может быть применён при создании генератора псевдослучайных чисел;
- ГОСТ Р 34.11-2012 («Стрибог») – описывается хеш-функция, которая может быть применена при создании генератора псевдослучайных чисел.

Сравнение КСГПСЧ:

HMAC-DRBG – реализован в mbedTLS и OpenSSL, считается достаточно криптографически защищенным. Есть теоретическое и «machine-checked» обоснование корректности.

HASH-DRBG – реализован в JCE FIPS, безопасность зависит от используемой хэш-функции. Работает быстро. Можно применять параллельные вычисления.

CTR-DRBG – можно использовать AES с большей длиной выхода, чем у 3DES, что дает преимущества безопасности. Можно применять параллельные вычисления.

ANSI X9.17 – широко используется (применяется в банковской сфере и PGP, может использовать IDEA вместо 3DES), есть некоторые теоретические уязвимости (Insertion attack).

ISAAC – не применяется широко и не стандартизирован, не исследован глубоко.

Российские государственные стандарты не предлагают конкретного алгоритма генерации стойких последовательностей случайных чисел. Однако предложены блочный шифр и хеш-функция, которые возможно использовать при реализации генератора. Основная проблема их использования заключается в недостаточной изученности перед публикацией.

По соотношению безопасности и скорости работы наиболее подходящим вариантом выглядит SHA-256 DRBG (см. Приложение 1). Блочные шифры работают значительно медленнее.

4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Псевдослучайные генераторы широко используются в современном мире:

- криптография;
- вероятностные алгоритмы и структуры данных;
- машинное обучение;
- повседневные задачи (например, определение победителей в розыгрышах);
- тестирование информационных систем (чтобы покрыть множество возможных сценариев, можно случайным образом генерировать входные данные).

4.2. Ориентировочная экономическая эффективность

В рамках данной курсовой работы расчет экономической эффективности не предусмотрен.

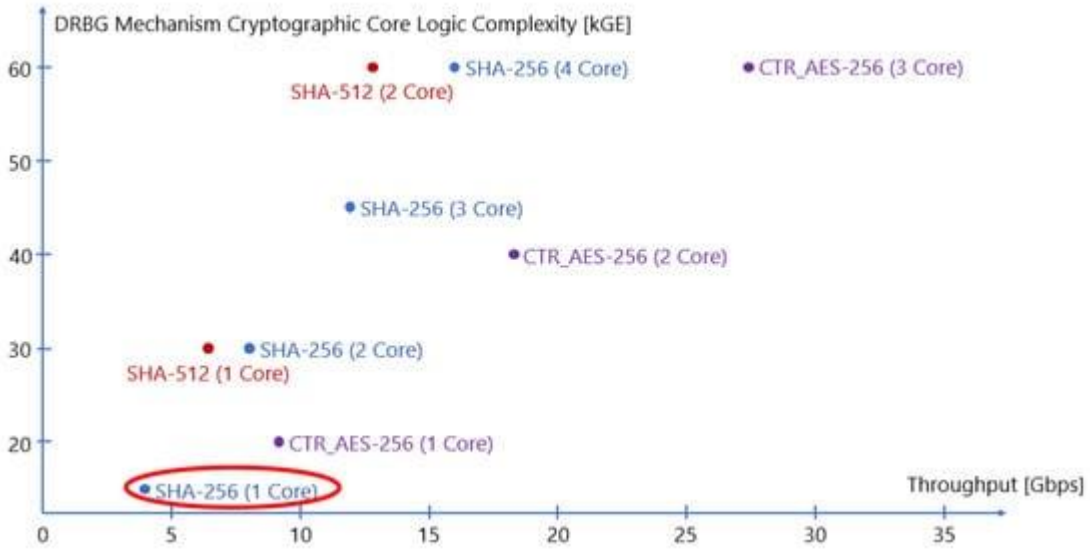
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Многие популярные генераторы псевдослучайных значений не являются криптографически стойкими. Например, всего 624 подряд идущих числа, сгенерированных известным генератором mt_19937, достаточно, чтобы предсказать все следующие числа, которые будет выдавать этот генератор. Реализуемый генератор будет криптографически стойким, при этом скорость его работы не будет значительно ниже аналогов, не являющихся криптографически стойкими (к примеру, mt_19937).

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) ГОСТ 19.102-77 Стадии разработки. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2) ГОСТ 19.101-77 Виды программ и программных документов. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3) ГОСТ 19.103-77 Обозначения программ и программных документов. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4) ГОСТ 19.104-78 Основные надписи. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5) ГОСТ 19.105-78 Общие требования к программным документам. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8) ГОСТ 19.603-78 Общие правила внесения изменений. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10) ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12) National Institute of Standards and Technology Special Publication 800-90A Revision 1. doi:10.6028/NIST.SP.800-90Ar1.
13) Введение в криптографию / Под общ. ред. В. В. Ященко. — 4-е изд., доп. М.: МЦНМО, 2012. — 348 с.
14) Документация MSDN [Электронный ресурс] URL https://docs.microsoft.com/

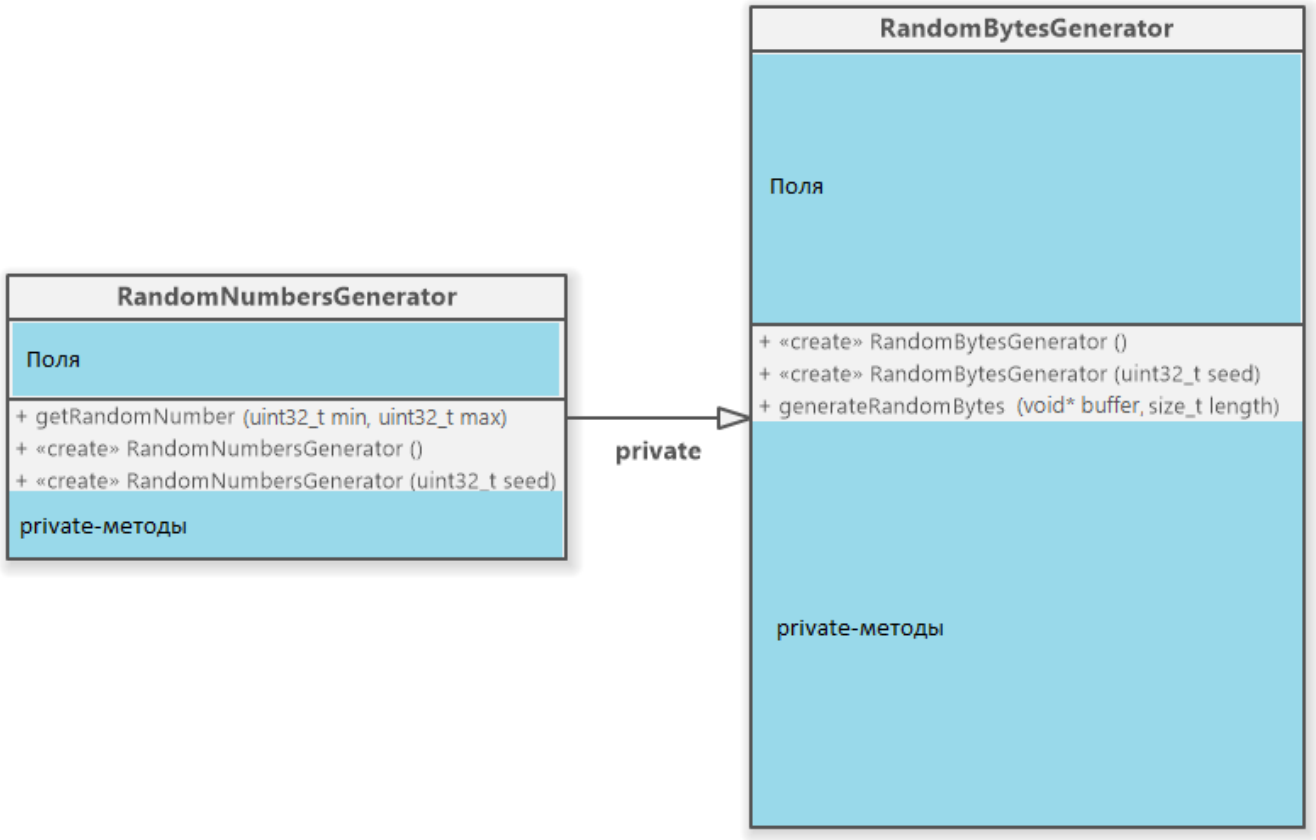
ПРИЛОЖЕНИЕ 1. ЭФФЕКТИВНОСТЬ КРИПТОСТОЙКИХ АЛГОРИТМОВ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ БИТ



ПРИЛОЖЕНИЕ 2. ДИАГРАММА И ОПИСАНИЕ КЛАССОВ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ И БИТ

Имя класса	Назначение	Дополнительные сведения
RandomBytesGenerator	Класс RandomBytesGenerator предназначен для генерации псевдослучайных последовательностей бит. Генератор предоставляет два варианта инициализации: – Инициализация системной энтропией – в таком случае генератор бит может быть использован для задач, связанных с безопасностью и криптографией (например, для генерации криптографических ключей). Выходную последовательность такого генератора практически невозможно предсказать; – Инициализация числом – такой генератор бит нужен, когда необходима возможность повторно получить сгенерированную псевдослучайную последовательность. Это может быть полезно, например, при тестировании – при нахождении зерна, последовательность, инициализированная которым, приводит к падению тестируемой системы, возможно заново воспроизвести проблемный сценарий.	При необходимости создания генератора данных другого формата возможно повторно использовать (наследование, композиция) RandomBytesGenerator для генерации случайных бит и преобразования их в другие данные (например, вещественные числа или более сложные объекты).
RandomNumbersGenerator	Класс RandomNumbersGenerator предназначен для генерации псевдослучайных чисел (с возможностью задания диапазона). Генератор предоставляет два варианта инициализации: – Инициализация системной энтропией – в таком случае генератор чисел может	RandomNumbersGenerator использует в своей основе класс RandomBytesGenerator (применяется private-наследование). Генерируемые псевдослучайные числа имеют равномерное распределение (также

	быть использован для задач, связанных с безопасностью и криптографией (например, для генерации криптографических ключей). Выходную последовательность такого генератора практически невозможно предсказать; – Инициализация числом – такой генератор чисел нужен, когда необходима возможность повторно получить сгенерированную псевдослучайную последовательность. Это может быть полезно, например, при тестировании – при нахождении зерна, последовательность, инициализированная которым, приводит к падению тестируемой системы, возможно заново воспроизвести проблемный сценарий.	при задании диапазона решена проблема «перекоса»).



Детали скрыты на основании соглашения о неразглашении результатов разработки (расписки NDA).

ПРИЛОЖЕНИЕ 3. ОПИСАНИЕ ПУБЛИЧНЫХ МЕТОДОВ И КОНСТРУКТОРОВ ПСЕВДОСЛУЧАЙНЫХ КЛАССОВ-ГЕНЕРАТОРОВ

Класс RandomBytesGenerator

Конструкторы

Спецификатор доступа	**Параметры конструктора**	**Назначение**
public	uint32_t seed	Генератор бит инициализируется «зерном» – несколько генераторов бит, инициализированных одинаковыми числами, будут выдавать одинаковые последовательности бит.
public		Генератор бит инициализируется системной энтропией – псевдослучайная последовательность бит такого генератора предсказать практически невозможно. Два разных генератора бит, инициализированных конструкторами без параметров, будут выдавать разные битовые последовательности.

Методы

Имя метода	**Спецификатор доступа**	**Тип значения, возвращаемого методом**	**Параметры метода**	**Назначение**
generateRandomBytes	public	void	void* buffer, size_t length	Генерирует в память по адресу buffer length псевдослучайных байт. Аналогичным образом происходит работа с генераторами псевдослучайных бит в популярных криптографических библиотеках.

Класс RandomNumbersGenerator

Конструкторы

Спецификатор доступа	**Параметры конструктора**	**Назначение**

public	uint32_t seed	Генератор чисел инициализируется «зерном» – несколько генераторов чисел, инициализированных одинаковыми числами, будут выдавать одинаковые числовые последовательности.
public		Генератор чисел инициализируется системной энтропией – псевдослучайная последовательность чисел такого генератора предсказать практически невозможно. Два разных генератора чисел, инициализированных конструкторами без параметров, будут выдавать разные последовательности.

Методы

Имя метода	**Спецификатор доступа**	**Тип значения, возвращаемого методом**	**Параметры метода**	**Назначение**
generateRandomNumber	public	uint32_t	uint32_t min, uint32_t max	Возвращает псевдослучайное число в диапазоне [min, max]. Генерируемые числа распределены равномерно.

Информация о приватных методах отсутствует на основании соглашения о неразглашении результатов разработки (расписки NDA).

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	Измененных	Заменен ных	Новых	Аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.10-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата