Ссылка на статью:
// Ученые записки УлГУ. Сер. Математика и информационные технологии. 2022, № 2, с. 108-116.

Поступила: 02.05.2022
Окончательный вариант: 21.06.2022

УДК 004.05

Разработка программного комплекса для массового статистического тестирования случайных последовательностей и алгоритмов шифрования

Смирнова А. В.

alinochka_gw@mail.ru
УлГУ, Ульяновск, Россия

В статье описано разработанное программное обеспечение для массового статистического тестирования случайных последовательностей и алгоритмов шифрования.

***Ключевые слова**: криптография, случайные последовательности, статистическое тестирование случайных последовательностей, NIST STS, программный комплекс.*

Введение

На этапах разработки и принятия алгоритмов шифрования в качестве стандартов важен анализ криптографических свойств алгоритма. Одним из этапов анализа криптографических свойств является статистическое тестирование.

В 1999 году был разработан пакет статистических тестов «NIST STS» в качестве инструмента для оценки генераторов псевдослучайных чисел (ГПСЧ) [7], предназначенных для использования в приложениях с использованием криптографии [1]. Статистические тесты NIST применялись также для анализа алгоритмов шифрования, представленных на конкурс AES [2]. В [6] указаны некоторые методики статистического тестирования алгоритмов шифрования, в основном они заключаются в создании ГПСЧ на основе исследуемого алгоритма шифрования.

В данной статье будет описан программный комплекс, основанный на генерации всевозможных шифртекстов на основе имеющихся входных сообщений, алгоритмов и режимов шифрования, ключей и векторов инициализации для последующего тестирования.

Также этот программный комплекс должен проводить статистическое тестирование всех исходных и получившихся битовых последовательностей.

Количество битовых последовательностей на одно входное сообщение вычисляется по формуле

$$n = 1 + n_k * (n_c * (1 + n_m * n_g)), \quad (1)$$

где n — количество битовых последовательностей, n_k — количество ключей, n_c — количество используемых алгоритмов шифрования, n_m — количество режимов шифрования, использующих вектор инициализации, n_g — количество векторов инициализации.

Для более тщательного анализа результатов статистического тестирования необходимо большое количество входных файлов, ключей и векторов инициализации, поэтому получающееся количество последовательностей значительно возрастает. Большая трудоемкость генерации зашифрованных последовательностей и статистического тестирования, а также большое количество исследуемых последовательностей оправдывает целесообразность использования распределенных вычислений.

1. Статистические тесты NIST

Статистические тесты возвращают одно или несколько P-значений, которые определяются как вероятность того, что идеальный генератор случайных чисел сгенерировал последовательность «менее случайную», чем исследуемая для статистики, определенной данным тестом. При P–значении, равном 1, считается, что последовательность является абсолютно случайной, а при P–значении, равном 0, считается, что последовательность абсолютно неслучайна. При этом обычно определяют уровень значимости a, который определяется как вероятность ошибки первого рода для нулевой гипотезы, в которой предполагается, что исследуемая последовательность является случайной. P–значение сравнивается с a, если P–значение превышает a, то принимается нулевая гипотеза, иначе нулевая гипотеза отклоняется, т.е. последовательность не является случайной. Например, значение a=0.01 означает, что из 100 случайных последовательностей не прошла бы тест только одна [1].

Статистические тесты NIST представляют собой 15 тестов, целью каждого из которых является выявление определенного недостатка исследуемой последовательности, дающего судить о том, что данная последовательность не является случайной. Некоторые из тестов подразделяются на несколько подтестов и возвращают несколько P–значений.

2. Пакет статистических тестов «NIST STS»

NIST STS представляет собой консольное приложение, предназначенное для статистического тестирования файлов и встроенных ГПСЧ. Данное приложение имеет следующие преимущества [1]:

- Кроссплатформенность;

- Расширяемость: пользователь может встроить в данное приложение новые тесты и ГПСЧ.

Недостатками для массового тестирования данного приложения являются:

- Текстовый формат вывода результатов, неудобный для последующей обработки программными средствами;
- Обработка последовательностей из единственного файла за один цикл работы;

3. Архитектура распределенной системы статистического тестирования

Распределенную систему статистического тестирования можно разделить на несколько подсистем: подсистема генерации зашифрованных последовательностей и подсистема статистического тестирования. Кроме того, для проведения массового тестирования необходимо средство для планирования заданий. Для этого была разработана подсистема учета новых файлов. Все три системы взаимодействуют с базой данных, куда записываются результаты тестирования и информация об исходных последовательностях.

Структура распределенной системы статистического тестирования показана на рис. 1.

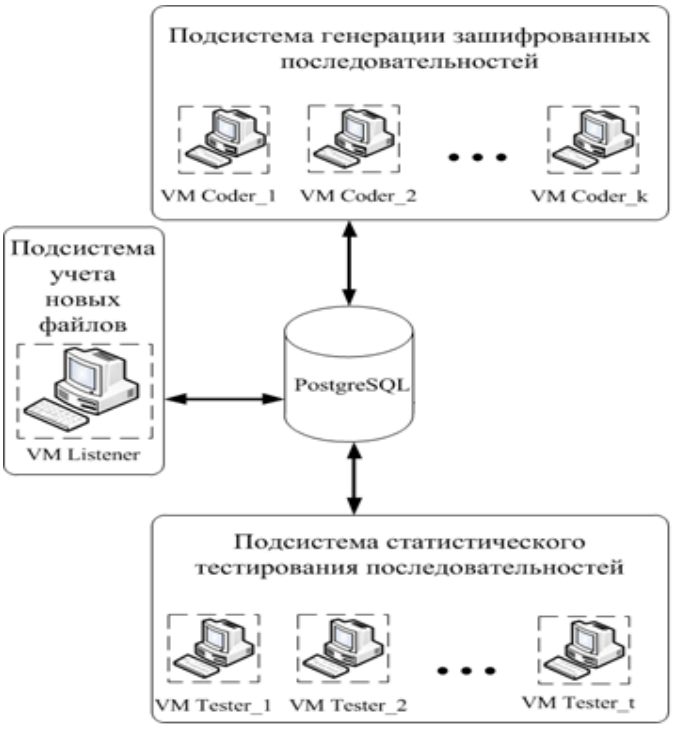

Рис. 1. Структура распределенной системы статистического тестирования.

На вход программного комплекса поступают следующие файлы: входные сообщения размера от 48 КБ до 6 МБ, ключи алгоритмов шифрования и векторы инициализации. Подсистема учета новых файлов записывает информацию об этих файлах в базу данных. По всем имеющимся входным сообщениям, алгоритмам и режимам шифрования, ключам,

и векторам инициализации генерируются новые зашифрованные последовательности подсистемой генерации зашифрованных последовательностей. Все входные и зашифрованные последовательности поступают на вход подсистеме статистического тестирования последовательностей, и результаты записываются в базу данных.

Все подсистемы реализованы как независимые программные модули. В целях обеспечения горизонтальной масштабируемости задачи генерации зашифрованных последовательностей и статистического тестирования могут быть распределены между несколькими виртуальными машинам одной локальной сети и на одном компьютере могут быть запущены несколько раз и работать одновременно. Схема проведения исследования с помощью данного программного комплекса показана на рис. 2.

Рис. 2. Схема проведения исследования.

Каждой последовательности соответствует одно задание. Задания генерируются программой учета новых файлов. Выполнение шифрования и тестирования контролируется кодами статусов заданий, он показаны в таблице 1.

Таблица 1. Коды статуса заданий

Код	Значение
1	Готово для шифрования
2	Готово для тестирования
3	Выполняется шифрование

4	Выполняется тестирование
5	Ошибка
6	Тестирование выполнено

Каждая последовательность в базе данных описывается идентификаторами следующих таблиц: исходный файл, ключ, алгоритм и режим шифрования, вектор инициализации.

4. Архитектура модуля учета новых файлов

Модуль учета новых файлов служит для генерации заданий при добавлении новых исходных файлов, таблиц замен, ключей и векторов инициализации, а также записи информации о новых файлах в базу данных.

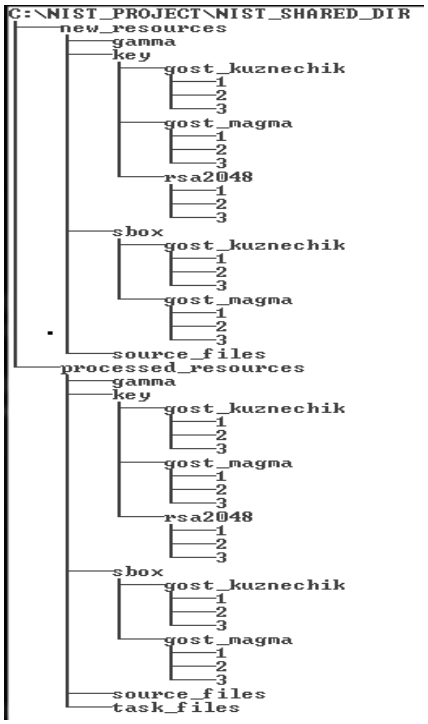

Рис. 3. Структура папки, используемой программным комплексом

Программа с некоторым периодом времени просматривает на наличие новых файлов общую папку, структура которой показана на рис. 3.

Структура подпапок new_resources и processed_resources практически идентична. В папке new_resources содержатся неучтенные данные, а в папке processed_resources — учтенные. По расположению нового файла программа идентифицирует категорию файла, начинает выполнять транзакцию, производит запись информации о файле и запрос для генерации новых заданий. При отсутствии ошибок результаты сохраняются, и файл перемещается к учтенным файлам своей категории.

Чтобы организовать тестирование файлов, необходимо выполнить следующие шаги:

1. Копировать входные сообщения, ключи шифрования и векторы инициализации в соответствующие подпапки папки new_resources;
2. Запустить программу учета новых файлов;
3. Запустить программы тестирования и программы шифрования.

Таким образом, участие пользователей сводится лишь к запуску программ и выбору файлов для тестирования.

5. Архитектура модуля генерации зашифрованных последовательностей

В модуле генерации зашифрованных последовательностей зашифрованные последовательности генерируются по информации, которая дана в строке таблицы task. В модуле

генерации зашифрованных последовательностей реализованы шифры RSA, ГОСТ 34.12—2015 «Кузнечик» и «Магма» [3] в режиме простой замены, в режиме сцепления блоков, в режиме распространяющегося сцепления блоков, в режиме обратной связи по шифртексту и в режиме обратной связи по выходу. Информация о режимах шифрования содержится в [4].

Для шифрования блока алгоритмом RSA используется реализация OpenSSL, в качестве ключей принимаются PEM–файлы. Алгоритмы шифрования ГОСТ 34.12—2015 «Кузнечик» и «Магма» были реализованы самостоятельно, в качестве ключей принимается бинарные файлы необходимого размера.

6. Архитектура модуля генерации зашифрованных последовательностей

Модуль статистического тестирования производит статистическое тестирование битовых последовательностей, путь к которым указан в таблице task. В программе тестирования реализовано 15 статистических тестов NIST [1]. Результатом статистического тестирования является вектор из 188 P–значений в таблице nist_results.

На вход программы могут быть приняты входные файлы размером от 48КБ до 6МБ. Программа с некоторым периодом времени запрашивает задания, у которых статус «Готово для тестирования» и путь к файлу. В разных потоках выполняются одновременно все тесты, кроме теста дискретного преобразования Фурье, теста на произвольные отклонения и вариант теста на произвольные отклонения — у этих тестов более высокие требования к оперативной памяти. Если тестирование выполнено успешно, то статус задания меняется на «Тестирование выполнено», результат тестирования записывается в таблицу nist_results.

7. Расширяемость функционала программного комплекса

В данном программном комплексе существует возможность расширения функционала.

При добавлении нового алгоритма шифрования необходимо добавление информации об алгоритме шифрования в базу данных, изменение структуры общей папки, изменение модуля учета новых файлов и изменение модуля генерации зашифрованных последовательностей.

При добавлении нового алгоритма тестирования необходимо изменение структуры таблиц результатов тестирования в базе данных и изменение модуля статистического тестирования.

На рис. 4 показаны схема изменений системы, которые необходимо осуществить для добавления новых алгоритмов шифрования и тестирования.

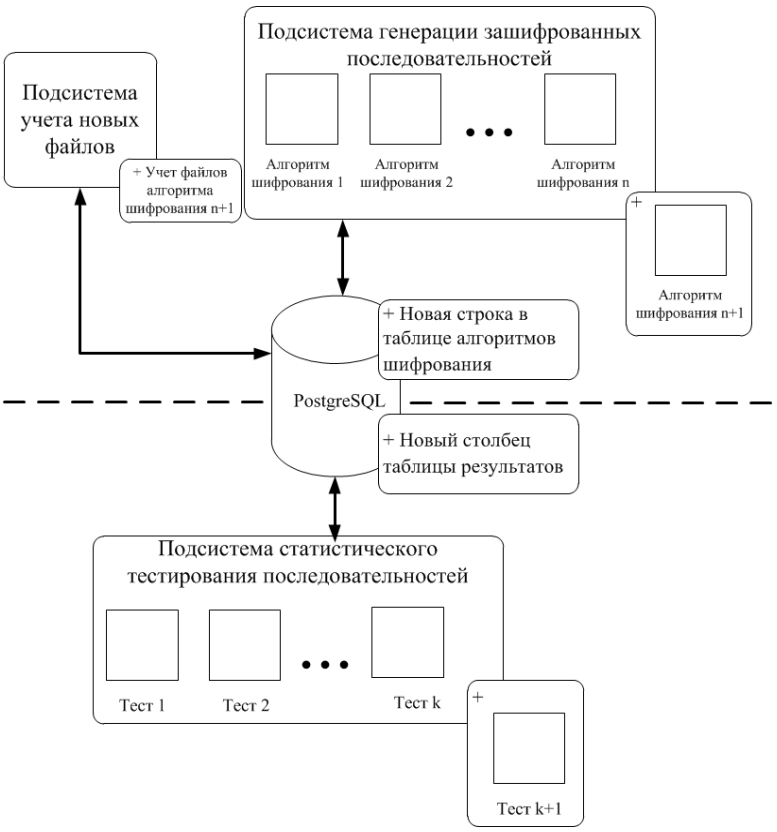

Рис. 4. Добавление новых алгоритмов тестирования и шифрования.

8. Описание программной реализации

Разработка программного комплекса велась на языке программирования *C++* на базе кроссплатформенного фреймворка Qt 5.13.1. Также при разработке использовались следующие внешние библиотеки:

- Boost-1.7.0 (неполные гамма-функции, логарифм гамма-функции, функция нормального распределения);
- OpenSSL-1.1.1c (хеш-функция SHA-1, алгоритм шифрования RSA).

По таким критериям, как отсутствие ограничения размера базы данных, устойчивость к нагрузкам была выбрана база данных PostgreSQL 12.4.

Программный комплекс состоит из трех консольных приложений, которые представляют собой соответствующие подсистемы и работают с базой данных определенной структуры, которая создается с помощью подготовленных скриптов SQL для данного программного комплекса. Информация, необходимая для подключения к созданной базе данных, записывается в конфигурационные файлы каждой из подсистем.

Полученный программный комплекс обладает следующими свойствами:

- кроссплатформенность,
- горизонтальная масштабируемость подсистемы генерации зашифрованной последовательности и подсистемы статистического тестирования,
- расширяемость программного комплекса,

- минимальное необходимое взаимодействие с пользователем,
- удобный формат вывода данных для последующей обработки результатов тестирования внешними программами.

В таблице 2 показаны временные характеристики статистического тестирования. Тестирование проводилось на компьютере со следующими параметрами: процессор 4 ядра Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz, оперативная память 4ГБ, SSD, Ubuntu 20.04 64-bit.

Таблица 2. Временные характеристики выполнения статистического тестирования

Количество	Минимальное время, с	Максимальное. время, с	Среднее время, с	Среднеквадратическое отклонение, с
1230200	0,178	336,694	2,431	7,719

Заключение

В данной статье была описана модель автоматизированной распределенной системы тестирования битовых последовательностей и ее программная реализация на языке программирования C++ на базе кроссплатформенного фреймворка Qt 5.13.1.

В будущем планируется применение данного программного на практике для сбора информации о зашифрованных последовательностях, также возможно исследование способов увеличения быстродействия программной реализации тестов. Одним из таких способов является использование параллельных вычислений [5].

Список литературы

1. Bassham L., Rukhin A., Soto J. , Nechvatal J., Smid M., Leigh S., Levenson M., Vangel M., Heckert N. and Banks D. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications // *Special Publication (NIST SP), National Institute of Standards and Technology*, Gaithersburg, MD, 2010. Режим доступа: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762 (дата обращения: 03.05.2022)
2. Soto J. and Bassham L. Randomness Testing of the Advanced Encryption Standard Finalist Candidates // *NIST Interagency/Internal Report (NISTIR)*, National Institute of Standards and Technology, Gaithersburg, MD, 2000 [online], https://doi.org/10.6028/NIST.IR.6483 (дата обращения: 03.05.2022)
3. ГОСТ Р 34.12-2015. *Информационная технология (ИТ). Криптографическая защита информации. Блочные шифры*.
4. ГОСТ Р 34.13-2015. *Информационная технология (ИТ). Криптографическая защита информации. Режимы работы блочных шифров.*

5. Вильданов Р. Р., Мещеряков Р. В., Бондарчук С. С. Тесты псевдослучайных последовательностей и реализующее их программное средство // *Доклады ТУСУР*. 2012, № 1(25), ч. 2, с. 108–111.
6. Ключарёв П.Г. О статистическом тестировании блочных шифров // *Математика и математическое моделирование*. 2018, № 5, с. 35-56. Режим доступа: https://doi.org/10.24108/mathm.0518.0000132 (дата обращения 03.05.2022).
7. Пикуза М.О., Михневич С.Ю. Тестирование аппаратного генератора случайных чисел при помощи набора статистических тестов NIST // *Доклады БГУИР*. 2021, т. 19, № 4, с.37-42. Режим доступа: https://doi.org/10.35596/1729-7648-2021-19-4-37-42 (дата обращения 03.05.2022).

Software package for mass statistical testing of random number sequences and cryptographic algorithms

Smirnova, A. V.

alinochka_gw@mail.ru

Ulyanovsk State University, Ulyanovsk, Russia

The paper describes software package implementation of NIST tests for mass statistical testing of random number sequences and cryptographic algorithms.

Keywords: *cryptography, random and pseudorandom number generators, statistical testing, NIST STS, software package.*