Разработка приложения для шифрования файлов

Лабораторная работа

Ревизия: 0.1

Последняя модификация: Vlad Kovtun

Дата последней модификации: 25.10.2009 20:46:00

История изменений

20.09.2009 – Версия 0.1. Первичный документ. Владислав Ковтун

Содержание

Лабораторная работа 3. Разработка приложения для шифрования файлов

Вопросы

1. Общие положения.

2. Постановка задачи.

3. Методические рекомендации.

Постановка задачи

Цель

1. Разработать консольное приложение, которое формирует ключ для симметричного алгоритма шифрования AES (Rijndael), а также производит зашифровывание и расшифровывание файлов посредством сгенерированного ключа.

2. Исследовать с помощью статистического теста NIST STS, зашифрованный файл в зависимости от исходного файла (который необходимо зашифровать).

Задачи

1. Разработать

2. Ознакомиться с теоретическими основами алгоритма шифрования AES (Rijndeal).

3. Ознакомиться с особенностями реализации алгоритма шифрования AES (Rijndeal) в криптопровайдере Windows.

4. Разработать консольное приложение для:

- Генерации ключа для алгоритма шифрования AES.
- Зашифровывания файлов алгоритмом AES.
- Расшифровывания файлов алгоритмом AES.

посредством криптопровайдера Windows.

5. Провести тестирование зашифрованного и исходного файлов посредством всех тестов пакета NIST STS.

6. Провести анализ результатов путем использования статистического критерия χ^2. Для этого:

- Построить гистограмму распределения частот, исходя из полученных результатов, и сравнить их с теоретическим распределением, используя критерий χ^2.
- Построить эмпирическую функцию распределения, сравнить ее с теоретическим распределением, пользуясь критерием Колмогорова.
- Сравнить таблицу результатов статистических исследований, в которой сравниваются результаты с эталонной выборкой BBS.

7. Подготовить и оформить отчет о проведенной лабораторной работе.

Вход

Для управления приложением предлагается использовать ключи командной строки:

- /c:<mode> – указывает, в каком режиме функционирует приложение. Допускаются следующие режимы:
    - g – режим формирования нового ключа.
    - e – режим зашифровывания данных.
    - d – режим расшифровывания данных.
- /m:<mode> - указывает режим шифрования и расшифровывания. В каждом режиме будет производится обработка данных:
    - CBC – The Cipher Block Chaining. Режим сцепления блоков.

    - CEB – The Cipher Feedback. Режим сцепления блоков по обратной связи.
    - CTS – The Cipher Text Stealing. Режим сокрытия текста. Аналогичен режиму ECB, за исключением двух последних блоков.
    - ECB – The Electronic Codebook. Режим электронной кодовой книги.
    - OFB – The Output Feedback. Режим сцепления блоков посредством обратной связи по выходу.
- /k:<filename> – полный путь к файлу, либо имени файла, который:
    - Может хранить вновь созданную ключевую последовательность.
    - Содержит существующий ключевую последовательность для режима расшифровывания и зашифровывания.
- /kl:<keylen> – указание длины ключа. Допускаются следующие длины ключей:
    - 128 бит.
    - 192 бита.
    - 256 бит.
- /i:<filename> - полный путь к файлу, либо имени файла, который содержит исходный файл. В зависимости от режима зашифровывания либо расшифровывания, указывается имя файла, который необходимо либо зашифровать, либо расшифровать, соответственно.
- /o:<filename> - полный путь к файлу, либо имени файла, который содержит результирующий файл. В зависимости от режима зашифровывания либо расшифровывания, указывается имя файла, который содержит либо зашифрованный файл, либо расшифрованный, соответственно.
- /? - вывод информации о допустимых ключах командной строки.

Пример

Генерация ключевой последовательности длиной 256 бит, которая заносится в файл key.dat.

C:/>aesecrypter.exe /c:g /k:key.dat /kl:256

Зашифровывание файла input.dat посредством ключа key.dat длиной 256 бит, результат зашифровывания заносится в файл encrypted.dat.

C:/>aesecrypter.exe /c:e /k:key.dat /kl:256 /i:input.dat /o:encrypted.dat

Расшифровыванием файла encrypted.dat посредством ключа key.dat длиной 256 бит, результат расшифровывания заносится в файл input.dat.

C:/>aesecrypter.exe /c:d /k:key.dat /kl:256 /i: encrypted.dat /o: input.dat

Вывод

Во время работы приложения, рекомендуется выводить информацию о статусе приложения, а также о корректности его работы на консоль.

Допускается перенаправление вывода консоли в текстовый файл, например:

C:/>aesecrypter.exe /c:g /k:key.dat /kl:256 >report.txt

Требования

Архитектура приложения строится по модульному принципу.

За основу принимается стандартная библиотека C++ (в случае разработки на C++) либо набор классов стандартной библиотеки C#.

Рекомендуется использовать защищенные ресурсы и указатели.

Алгоритм

Кратко опишем алгоритм данного приложения:

1. Проверка корректности параметров командной строки.

2. Выбор режима функционирования приложения: генерация ключа, зашифровывание, расшифровывание.

3. Генерация ключа.

3.1. Формирование ключевого контейнера.

3.2. Запрос пароля у пользователя для доступа к ключевому контейнеру. Подразумевается повторный ввод пароля, для его подтверждения. Сформированный ключ следует зашифровать ключом, сформированным посредством хеш-преобразования над введенным паролем.

3.3. Генерация ключа.

3.4. Открытие файла для сохранения ключевого контейнера. Проверка на корректность.

3.5. Запись ключевого контейнера. Проверка на корректность.

3.6. Закрытие файла содержащего ключевой контейнер.

4. Зашифровывание.

4.1. Открытие файла содержащего ключевой контейнер. Проверка на корректность.

4.2. Чтение ключевого контейнера из файла. Проверка на корректность.

4.3. Закрытие файла ключевого контейнера.

4.4. Запрос у пользователя пароля, для доступа к ключу.

4.5. Открытие файла с исходными данными. Проверка на корректность.

4.6. Открытие файла с результирующими данными (зашифрованными). Проверка на корректность.

4.7. Чтение блока данных, выполнение над ним процедуры зашифровывания.

4.8. Запись зашифрованного блока данных. Проверка на корректность.

4.9. Закрытие файла с исходными данными. Проверка на корректность.

4.10. Закрытие файла с результирующими данными. Проверка на корректность.

5. Расшифровывание.

5.1. Открытие файла содержащего ключевой контейнер. Проверка на корректность.

5.2. Чтение ключевого контейнера из файла. Проверка на корректность.

5.3. Закрытие файла ключевого контейнера.

5.4. Запрос у пользователя пароля, для доступа к ключу.

5.5. Открытие файла с зашифрованными данными. Проверка на корректность.

5.6. Открытие файла с результирующими данными (расшифрованными). Проверка на корректность.

5.7. Чтение блока зашифрованных данных, выполнение над ним процедуры расшифровывания.

5.8. Запись расшифрованного блока данных. Проверка на корректность.

5.9. Закрытие файла с исходными (зашифрованными) данными. Проверка на корректность.

5.10. Закрытие файла с результирующими (расшифрованными) данными. Проверка на корректность.

Порядок выполнения работы

1. Разработать консольное Windows приложение для: генерации ключей, зашифровывания и расшифровывания файлов.

2. Зашифровать файл размером 12 Мб.

3. Запустить приложение NIST_STS.exe для статистического тестирования.

3.1. Выполнить статистическое тестирование заданных зашифрованного и расшифрованного файлов. Для этого необходимо воспользоваться руководством Приложение А.

Используя обобщенные результаты тестирования, которые находятся в файле finalAnalysisReport, выполнить интерпретацию результатов тестирования (См. пример в Приложение Б).

Используя результаты тестирования из файла finalAnalysisReport, для указанных в работе статистических тестов построить теоретическую и эмпирическую гистограмму частостей F_k попадания значений P_{ij} в каждый из интервалов $k=\overline{1,10}$ подинтервалов, на которые разбит интервал $[0,1]$, рис. 1.

Согласно выражения:

$$\chi_j^2 = \sum_{k=1}^{10} \frac{\left(F_k - m/10\right)^2}{m/10}$$

Рассчитать значение χ^2 и проверить согласованность теоретического и эмпирического закона распределения значений P_{ij} при $\alpha = 0{,}01$ и $\alpha = 0{,}05$. Критические значения для χ^2 представлены в Приложении В.

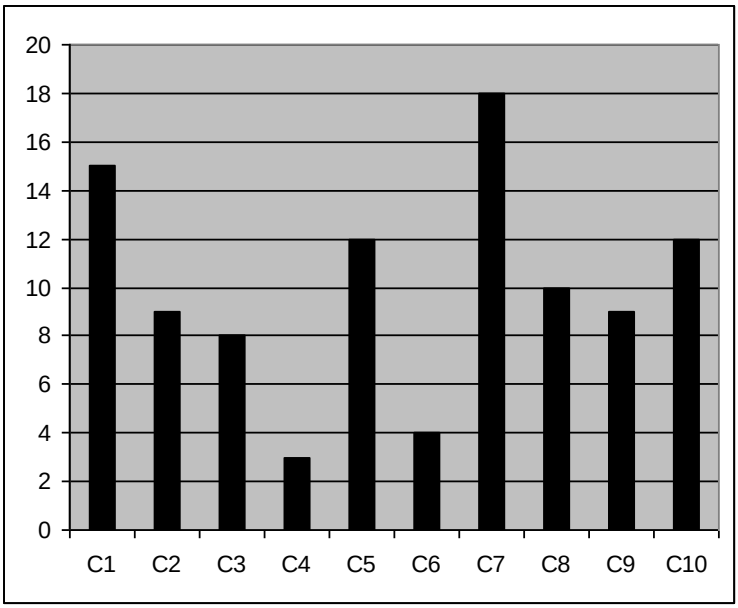

Рис. 1. Образец построения гистограммы

Проверить согласованность эмпирического и теоретического закона распределения значений P_{ij} для указанных в работе статистических тестов посредством критерия Колмогорова-Смирнова. Для этого, посредством результатов тестирования, что находятся в файле result, соответствующего теста, построить эмпирическую функцию распределения и теоретическую функцию распределения для равновероятного закона. Вычислить статистику Колмогорова-Смирнова, см. Приложение В. Как критерий оценки разницы оценки теоретической $F_t(P_{ij})$ и эмпиричной $F_e(P_{ij})$ функции распределения рассматривается максимальное значение модуля разницы:

$$D = \max\left|F_T(P_{ij}) - F_e(P_{ij})\right|.$$

Определяется значение:

$$\beta = D\sqrt{m},$$

где m – количество точек, по которым строится эмпирическая кривая.

Зная β, представлено в таблице 2 Приложения В вычислить вероятность $P(\beta)$. Если эта вероятность небольшая, то гипотеза о согласованности законов отклоняется, м наоборот, если вероятность большая, то гипотеза принимается.

Проверить согласованность результатов, которые были получены с помощью критерия χ^2 и критерия Колмогорова-Смирнова, сделать выводы.

Замечание: С целью упрощения заполнения отчета, следует пользоваться электронными таблицами Microsoft Excel.

Теоретические сведения

Детальную информацию о работе шифра AES (Rijndeal) можно найти в лекции 3. Далее остановимся на особенностях реализации консольного Windows приложения посредством языка высокого уровня C#, на базе Microsoft .NET Framework 3.0.

Начиная с Microsoft .NET Framework 2.0 появилась реализация алгоритмов блочного шифрования, так для реализации шифра AES, используется класс `RijndealManaged`. Режимы работы данного алгоритма описаны в свойстве `Mode` посредством перечисления `CipherMode`. Более детальную информацию касательно написания приложения и примеров, можно почерпнуть в описании методов класса `RijndealManaged`.

Литература

1. Microsoft Developer Network. Доступно по адресу: www.msdn.com

2. Теоретические материалы и программное обеспечение доступны по адресу: www.nrjetix.com/r-and-d/lectures

3. Руководство пользователя к пакету NIST STS. Доступно: NIST STS Guide

4. Средство для статистического анализа случайных последовательностей. Доступно: NIST STS Software

5. Потій О.В., Лєншин А.В., Горбенко Ю.І. Методичні вказівки до лабораторних робіт за дисципліною «стандартизація та сертифікація в галузі інформаційної безпеки». ХНУРЕ. -2005. –С. 105.

Приложение А. Руководство исследователя к тесту NIST STS

Средством тестирования NIST STS следует пользоваться из командной строки. Для этого необходимо произвести запуск исполняемого файла NIST_STS.exe, и длины исследуемой последовательности, например, 12 582 912 (рекомендуемая длина 1 000 000), см. рис. 2.

Рис. 2. Запуск приложения NISTSTS.exe

После запуска приложения необходимо указать, какие данные необходимо тестировать, «OPTION---->»:

- Тестовые данные будут сформированы генератором, реализованным в тесте NISTSTS.exe. Для этого необходимо ввести значение от 1 до 9. Например вводим

9 для выбора генератора основанного на функции хеширования SHA-1 («G Using SHA-1»). После чего будет сформирована последовательность длиной 12 Мб.

- Тестовые данные находятся в файле, который содержит последовательность, которую необходимо протестировать. Для этого необходимо указать значение 0. В строке «User Prescribed Input File: » вводится имя файла, например «output.bin», Рис. 2.

После чего будет предоставлен перечень статистических тестов, которые могут быть применены к тестируемой последовательности. Если исследователь не желает использовать существующие тесты, ему следует ввести в строке «Enter Choise: » цифру 0, в противном случае 1, рис. 3 и нажать клавишу «Enter».

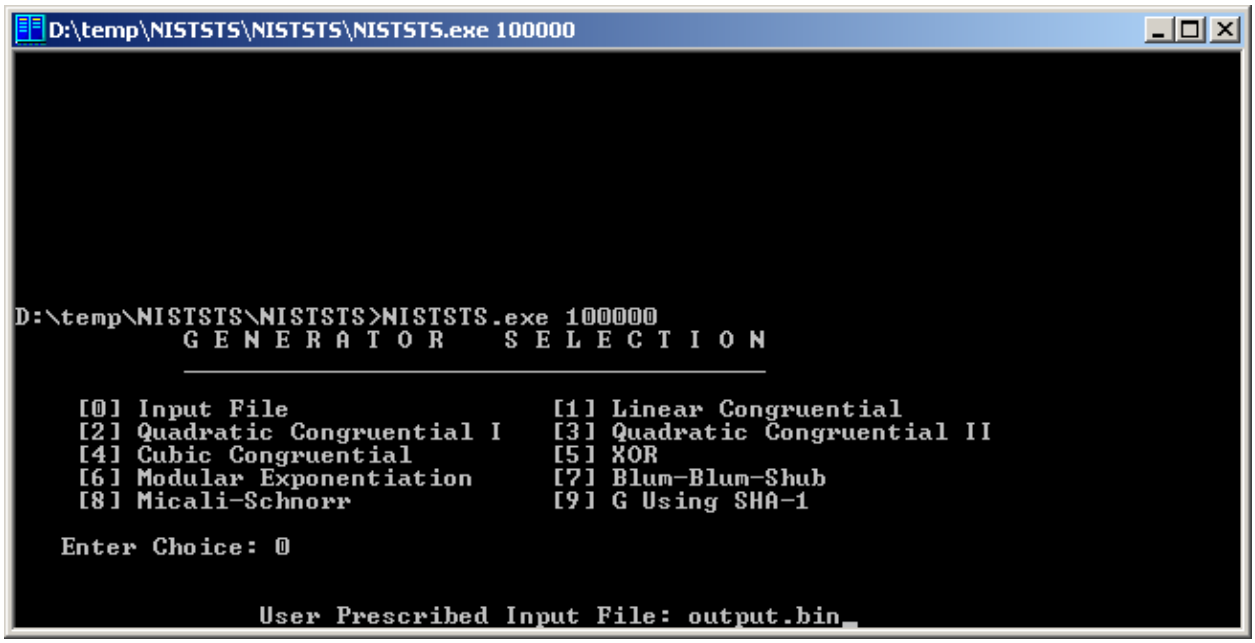

Рис. 3. Выбор тестируемой последовательности

Далее программа предложит выбрать, какие именно тесты следует применить к последовательности. Все присутствующие тесты пронумерованы от 01 до 15. Исследователю следует ввести в нижней строке под номером соответствующего теста цифру 1, если тест следует применить, в противном случае – цифру 0, рис. 4.

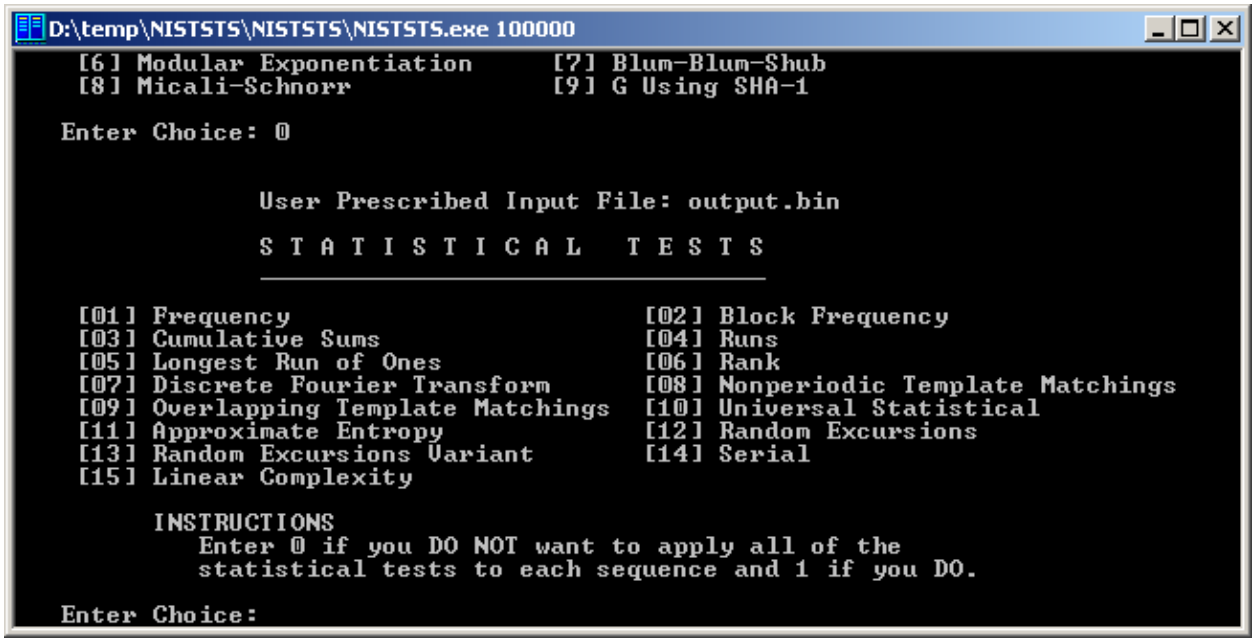

Рис. 4. Выбор единственного статистического теста

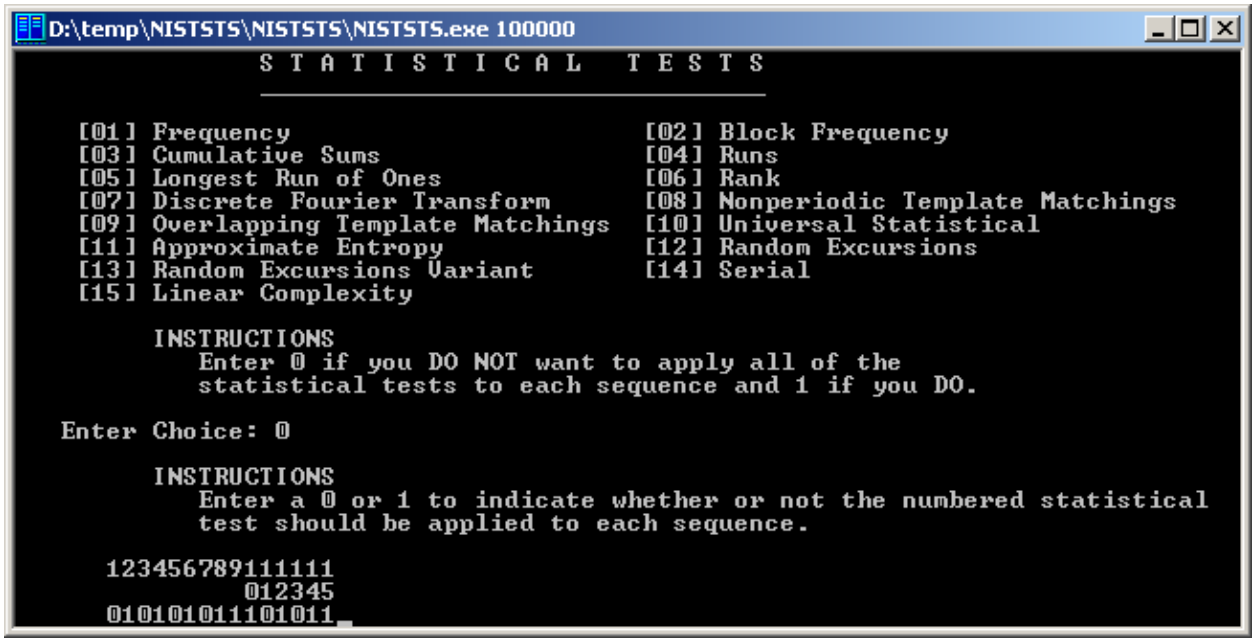

Рис. 5. Выбор множества статистических тестов, которые будут последовательно применяться к исследуемой последовательности

После выбора необходимых тестов, приложение предложит ввести параметры, согласно которым будет производиться тестирование последовательности, количество тестируемых последовательностей, и указать в каком формате будут представлены данные в файле:

- «текстовый» – один символ – один бит;
- «двоичный» - один байт содержит 8 бит последовательности.

В строке «How many bitstreams should be generated?» исследователю следует ввести количество тестируемых последовательностей. NIST STS рекомендует число 100 в качестве количества посдпоследовательностей.

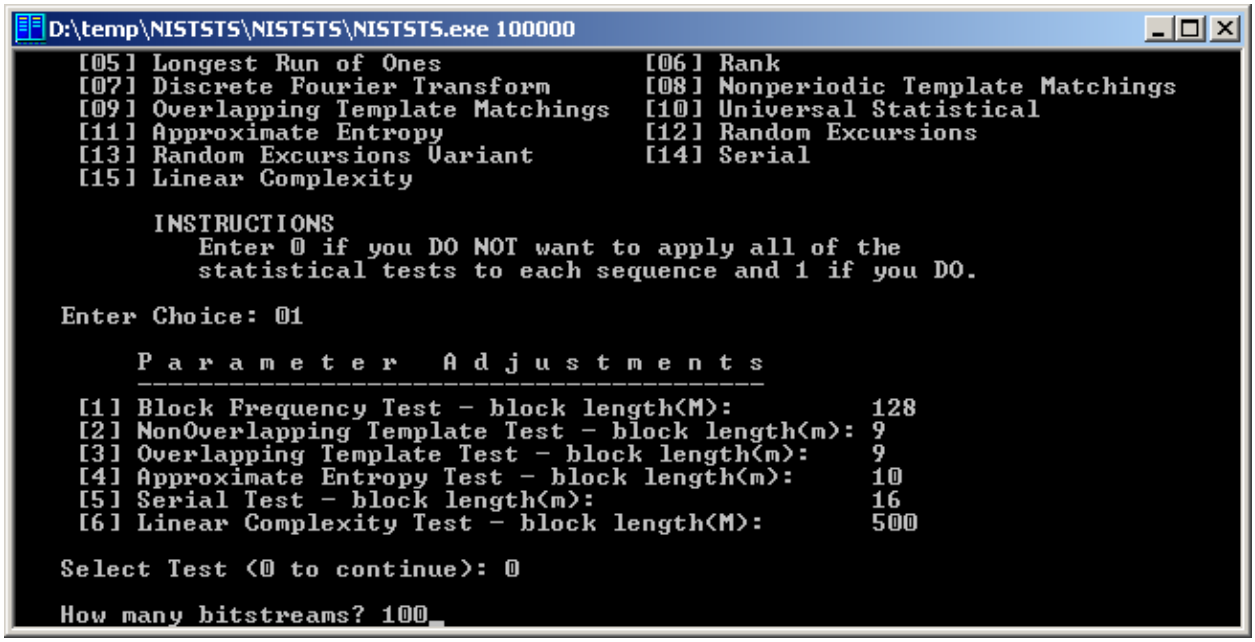

Рис. 6. Настройка параметров Частотного теста, например

В строке «Select input mode:» исследователь указывает тип тестируемой подпоследовательности: в случае если последовательность представлена в ASCII формате, следует ввести 0, если в двоичном, то 1, рис. 7.

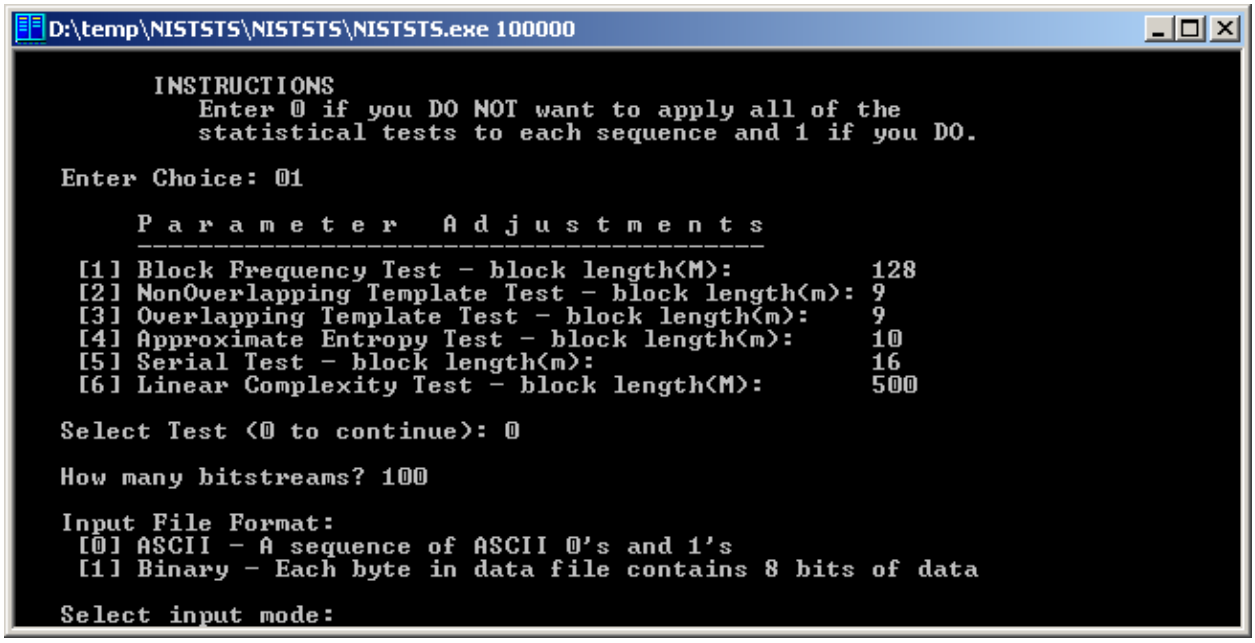

Рис. 7. Выбор формата представления последовательности

После ввода типу последовательности приложение выведет сообщение «Statistical Testing In Progress…….», рис. 8.

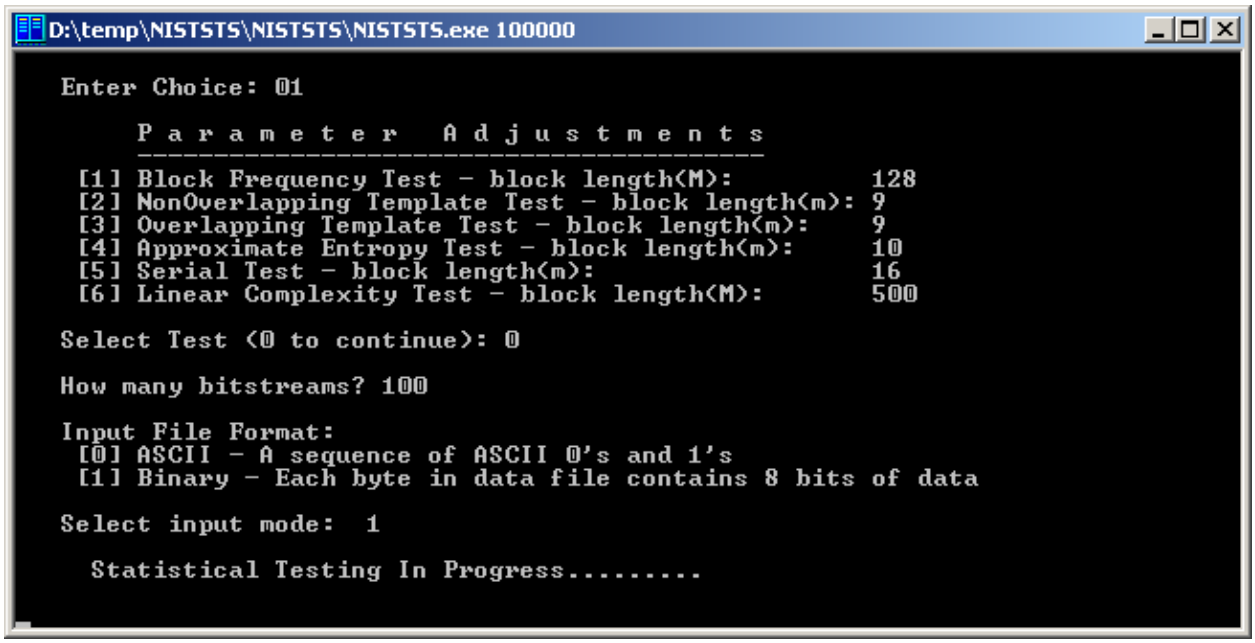

Рис. 8. Отображение статуса теста

После завершения работы приложения, все суммарные расчетные данные размещаются в том же каталоге, где находится само приложение, в файле finalAnalysisReport, рис. 9 и рис. 10.

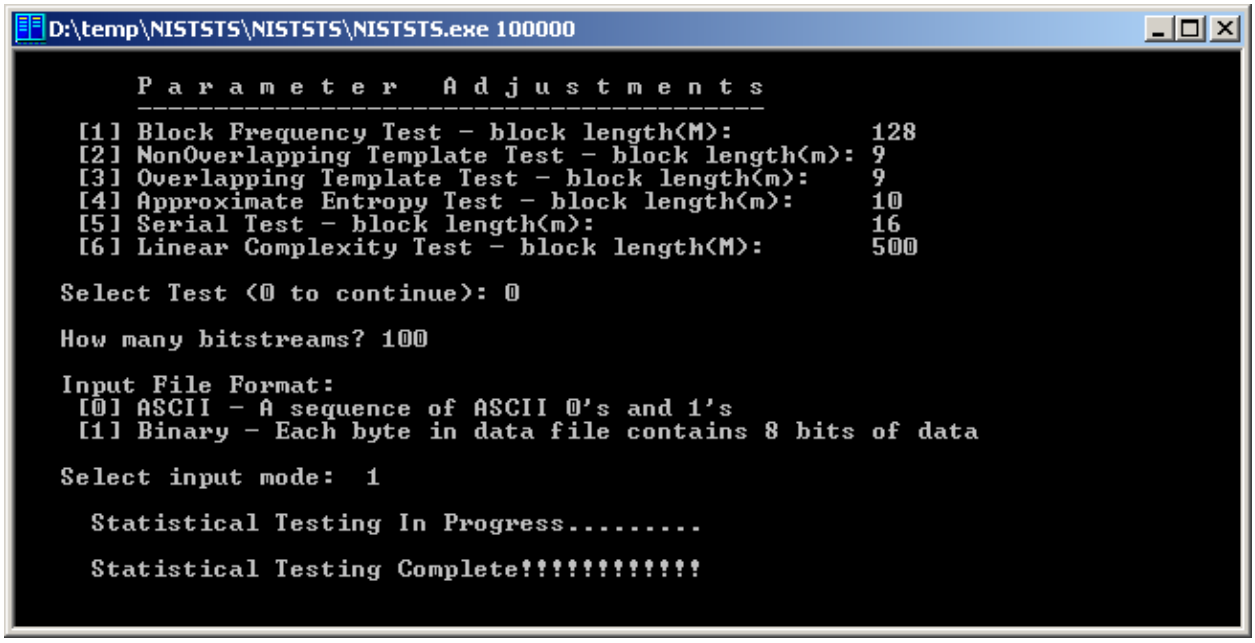

Рис. 9. Сообщение о завершении теста

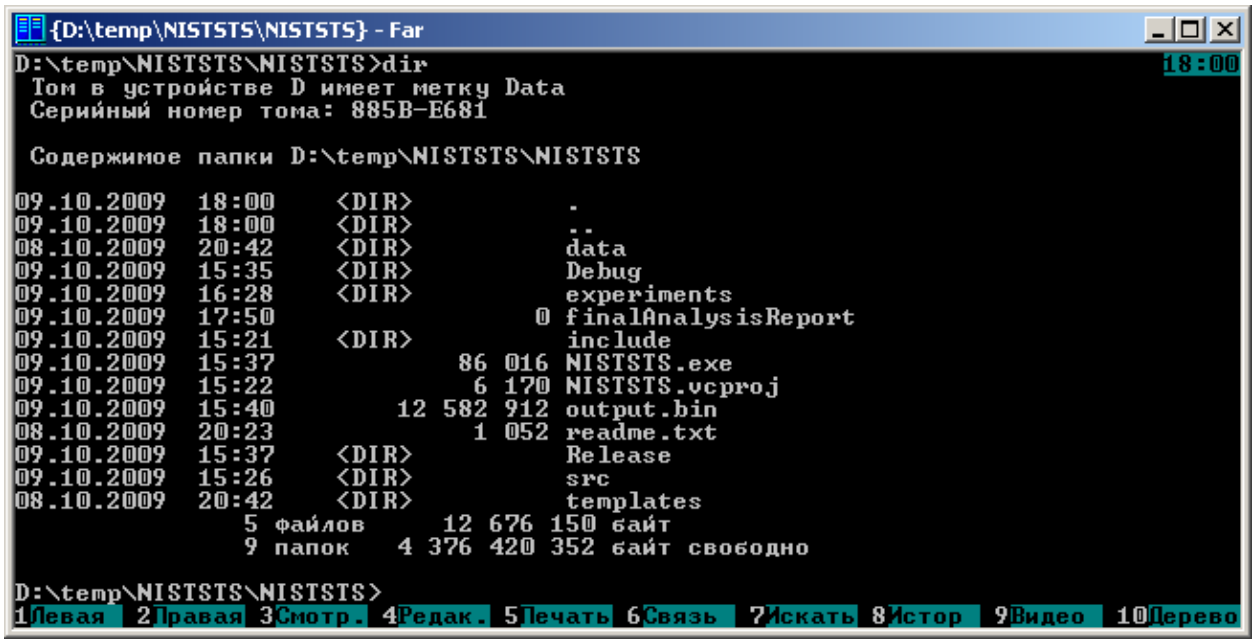

Рис. 10. Информация о результатах тестирования

Приложение Б. Пример интерпретации результатов теста генератора

Випробування здійснювалися у спеціальній лабораторії ЗАТ «Інститут інформаційних технологій» відповідно до методики тестування NIST SP 800-22. Випробування здійснювалися представниками на випробувальному стенді.

Інтерпретація результатів перевірки програмної реалізації алгоритму блокового шифрування FIPS 197 у режимі лічильника

Програмна реалізації алгоритму блокового шифрування FIPS 197 у режимі лічильника була піддана статистичному тестуванню з використанням методики NIST STS, що рекомендована Національним інститутом зі стандартизації і технологій США - NIST SP 800-22.

З використанням методики NIST STS було здійснене тестування програмної реалізації алгоритму блокового шифрування FIPS 197 у режимі лічильника, а також, з метою порівняльного аналізу, генератора псевдовипадкових чисел BBS (тестова вибірка, рекомендована NIST).

Для здійснення тестувань були обрані такі параметри:

- довжина послідовності, що тестується $n = 10^6$ біт;
- кількість послідовностей, що тестується $m = 100$;
- рівень значущості $\alpha = 0{,}01$.

Таким чином, обсяг вибірки, що тестується, склав $N = 10^6 \times 100 = 10^8$ біт;

- кількість тестів (q) для різних довжин $q = 189$, таким чином, статистичний портрет генератора містить 18900 значень імовірності *P.*

В ідеальному випадку при $m = 100$ і $\alpha = 0{,}01$ у ході тестування може бути відкинута тільки одна послідовність зі ста, тобто коефіцієнт проходження кожного тесту має складати 99%. Але це занадто жорстке правило. Тому застосовується правило на основі довірчого інтервалу для r_j. Нижня межа в цьому випадку складе значення $r_{min} = 0{,}96015$. З цих позицій аналізуються результати тестування генераторів.

У таблиці Б.1 наведено результати проходження тестування програмної реалізації алгоритму блокового шифрування FIPS 197 у режимі лічильника за Правилом 1.

Всі тести пройшли (або вказати який тест не пройшов, та у відповідності до керівництва NIST STS надати інтерпретацію недоліку, що виявляється цім тестом)

У таблиці Б.2 наведено результати проходження тестування програмної реалізації алгоритму блокового шифрування FIPS 197 у режимі лічильника за Правилом 2.

Таблиця В.1– Результати тестування алгоритму FIPS 197 за правилом 1

Генератор	Кількість тестів, у яких тестування пройшли більш 99% послідовностей	Кількість тестів, у яких тестування пройшли більш 96% послідовностей
BBS	134 (71%)	189 (100%)
FIPS 197	126 (67%)	189 (100%)

Таблиця В.2 – Результати тестування алгоритму FIPS 197 за правилом 2

Генератор	Кількість тестів, у яких значення *P < 0,001*	Кількість тестів, у яких значення *P < 0,01*
BBS	134 (71%)	189 (100%)
FIPS 197	126 (67%)	189 (100%)

(Якщо необхідно, вказати за якими тестами значення *P* було нижче визначеної межі, вказати, які недоліки виявляють ці тести, та порівняти ці результати з результатами тестування за Правилом 1.)

У Додатку В наведено результати обчислення, а також побудовано статистичний портрет.

Таким чином, можна зробити висновок, що програмна реалізація алгоритму блокового шифрування FIPS 197 у режимі лічильника пройшла комплексний контроль за методикою NIST STS.

Приложение В. Критерии согласованности

При побудові ключових систем одним з основних завдань є одержання випадкових і псевдовипадкових послідовностей, які не можуть бути відрізнені від випадкових і мають великий період. Після генерації послідовності чисел $X = \{x_1, x_2, \ldots, x_n\}$ необхідно впевнитись у тому, що випадкова величина X має рівномірний закон розподілу, її реалізації випадкові та незалежні. Методи математичної статистики дають нам можливість побудувати статистичні тести для перевірки гіпотез про рівномірність, випадковість і незалежність випадкових величин.

Для перевірки гіпотези про закон розподілу скористаємося критеріями χ^2 Пірсона та критерієм Колмогорова-Смірнова.

Критерій Пірсона χ^2

Критерій Пірсона χ^2. перевіряє узгодженість гіпотетичних ймовірностей $P_k = P(x_k)$ випадкових величин x_1, x_2, …, x_n з їхніми відносними частотами $h_k = v_k / n$ у вибірці з n незалежними спостереженнями. Статистика критерію має вигляд:

$$\chi^2 = n\sum_{k=1}^{m} \frac{(h_k - P_k)^2}{P_k},$$

де m – кількість інтервалів розбивки. Граничне значення статистики для рівня значущості визначається за формулою:

$$\chi_\alpha^2 \approx l\left(1 - \frac{2}{9l} + z_\alpha \sqrt{\frac{2}{9l}}\right)^3,$$

де l – кількість ступенів волі; z_α - граничне значення стандартного нормального розподілу.

Гіпотеза про узгодженість емпіричного закону розподілу спостережуваної випадкової величини з теоретичним відкидається, якщо спостережене значення $\chi^2 > \chi_\alpha^2$.

Критерій Колмогорова-Смірнова

Критерій Колмогорова-Смірнова заснований на розподілі величини

$$D_n = \max\left|F_n(x) - F(x)\right|,$$

де $F_n(x)$ – емпіричний закон розподілу; $F(x)$ – гіпотетичний закон розподілу. Відомо, що яка б не була безперервна функція розподілу $F(x)$, імовірність

$$P\left\{D_n < \frac{\lambda}{\sqrt{n}}\right\}$$

при $n \to \infty$ прагне до межі: $K(\lambda) = 1 - 2\sum_{k=1}^{\infty} (-1)^{k-1} e^{-2\lambda^2 k^2}$.

Таблиці В.1– Деякі дані для розподілення χ^2

	p =0,99	p =0,95	p =0,75	p =0,50	p =0,25	p =0,05	p =0,01
ν =1	0,00016	0,00393	0,1015	0,4549	1,323	3,841	6,635
ν =2	0,00201	0,1026	0,5753	1,386	2,773	5,991	9,210
ν =3	0,1148	0,3518	1,213	2,366	4,108	7,815	11,34
ν =4	0,2971	0,7107	1,923	3,357	5,385	9,488	13,28
ν =5	0,5543	1,1455	2,675	4,351	6,626	11,07	15,09
ν =6	0,8720	1,635	3,455	5,348	7,841	12,59	16,81
ν =7	1,239	2,167	4,225	6,346	9,037	14,07	18,48
ν =8	1,646	2,733	5,071	7,344	10,22	15,51	20,09
ν =9	2,088	3,325	5,889	8,343	11,39	16,92	21,67
ν =10	2,558	3,940	6,737	9,342	12,55	18,31	23,21

	p =0,99	p =0,95	p =0,75	p =0,50	p =0,25	p =0,05	p =0,01
ν =11	3,053	4,575	7,584	10,34	13,70	19,68	24,73
ν =12	3,571	5,226	8,438	11,34	14,84	21,03	26,22
ν =15	5,229	7,261	11,04	14,34	18,25	25,00	30,58
ν =20	8,260	10,85	15,45	19,34	23,85	31,41	37,57
ν =30	14,95	18,49	24,48	29,34	34,80	43,77	50,89
ν =50	29,71	34,76	42,94	49,33	56,33	67,50	76,15
ν >30				Приблизно $\nu + 2\sqrt{\nu \cdot x}_p + \frac{4}{3}x_p^2 - \frac{2}{3}$			
x_p	−2,33	−1,64	−0,675	0,00	0,675	1,64	2,33

Для практичних обчислень слід застосовувати такі формули:

$$D_n^+ = \max_{1\le m\le n}\left(\frac{m}{n} - F(\xi_m)\right),$$

$$D_n^- = \max_{1\le m\le n}\left(F(\xi_m) - \frac{m-1}{n}\right),$$

$$D_n = \max\left(D_n^+, D_n^-\right),$$

де $\xi_1 \le \xi_2 \le \ldots \le \xi_n$ – упорядковані значення випадкової величини.

Статистика D_n має певний розподіл (розподіл Колмогорова), що має табульований для деяких значень n. При $n \ge 10$ для визначення граничного значення $D_n(\alpha)$ на відрізку $0{,}01 \le \alpha \le 0{,}2$ потрібно користуватися формулою

$$D_n(\alpha) = \sqrt{\frac{1}{2n}(y) - \frac{2y^2 - 4y - 1}{18n}} - \frac{1}{6n} \approx \sqrt{\frac{y}{2n}} - \frac{1}{6n},\ y = -\ln(0{,}5\alpha).$$

При $n \ge 100$ зазначена формула правильна для всіх $0{,}0001 \le \alpha \le 0{,}5$.

Якщо в результаті досвіду виявиться, що $D_n \ge D_n(\alpha)$ то гіпотезу про узгодженість емпіричного та гіпотетичного законів розподілу слід відкинути з рівнем значущості α.

Таблиця В.2– Деякі дані для обчислення критерію Колмогорова-Смірнова

β	$P(\beta)$	β	$P(\beta)$	β	$P(\beta)$	β	$P(\beta)$	β	$P(\beta)$	β	$P(\beta)$
0	1,0	0,3	1,0	0,6	0,654	0,9	0,393	1,2	0,112	1,5	0,022
0,1	1,0	0,4	0,997	0,7	0,711	1,0	0,270	1,3	0,068	1,6	0,012
0,2	1,0	0,5	0,994	0,8	0,544	1,1	0,179	1,4	0,040	1,7	0,006