Лабораторна робота №12

Вивчення можливостей статистичного пакету NIST STS.

Мета: навчитися статистичному аналізу результатів роботи криптографічних примітивів за допомогою пакету статистичного тестування NIST STS.
Обладнання: персональний комп'ютер.
Програмне забезпечення: операційна система Windows (не старша за Windows 7); пакет статистичного тестування NIST STS (NIST Statistical Suite) v. 2.1.2; будь-яка мова програмування; криптофреймворк (криптобібліотека), де реалізовано основні криптоалгоритми.

Теоретична частина

Як визначити, наскільки якісний шифр або генератор псевдовипадкових послідовностей ми розробили? За якими критеріями це можна зробити?
Однією з основних характеристик будь-якого криптографічного програмного забезпечення (блокового шифру, генератора псевдовипадкових послідовностей тощо) є його статистичні характеристики. Результат шифрування/генерування випадкової послідовності повинен мінімально відрізнятися від випадкового числа.
Для визначення цієї різниці криптографічна спільнота використовує пакет статистичного тестування, розроблений NIST перед конкурсом AES для тестування генераторів псевдовипадкових послідовностей. З того часу він став стандартом для оцінки статистичних характеристик шифрів, генераторів та інших криптографічних примітивів.
Пакет містить 16 статистичних тестів. В залежності від вхідних параметрів обчислюється 189 значень імовірності *P*, які можна розглядати як результат роботи окремих тестів. У табл. 1 приводяться зібрані дані по усіх тестах із вказівкою кількості значень, що обчислюються, імовірності *P*, фізичного змісту статистики тесту і дефекту, на виявлення якого спрямовано тест.

Таблиця 1

Опис статистичних тестів пакету NIST STS

№ з/п	Статистичний тест	Статистика тесту c(S)	Дефект, що виявляється
1	Частотний (монобітний тест)	Нормалізована абсолютна сума значень елементів послідовності	Надто багато нулів або одиниць у послідовності
2	Частотний тест (в середині блоку)	Міра узгодженості кількості одиниць, що спостерігаються із тим, що очікується теоретично.	Локалізовані відхилення частоти появи одиниць в блоці від ідеального значення ½
3	Перевірка накопичених сум	Максимальне відхилення значень накопиченої суми елементів послідовності від початкової точки відліку (точка 0)	Велика кількість одиниць або нулів на початку або наприкінці двійкової послідовності
4	Перевірка серій	Загальна кількість серій на усій довжині послідовності	Надто швидка або надто повільна зміна знака у ході генерації послідовності
5	Перевірка максимальної	Міра узгодженості значень максимальної довжини, що	Відхилення від теоретичного закону розподілення

	довжини серії у блоці.	спостерігаються, із значенням, що очікується теоретично	максимальних довжин серій одиниць.
6	Перевірка рангу двійкової матриці	Міра узгодженості значення рангів різного порядку, що спостерігаються, із значенням, що очікується теоретично	Відхилення емпіричного закону розподілення значень рангів матриць від теоретичного, що вказує на залежність символів у послідовності.
7	Спектральний аналіз на основі дискретного перетворення Фур‘є	Нормалізована різниця кількості частотних компонент, що спостерігаються із тією, що очікується, які перевищують 95% рівень порогу.	Виявлення періодичних складових (трендів) у двійковій послідовності.
8	Перевірка шаблонів, що перекриваються	Міра узгодженості кількості шаблонів, що перекриваються, у послідовності із теоретичним значенням.	Велика кількість m- бітних серій із одиниць у послідовності.
9	Універсальний тест Маурера	Сума логарифму відстані між l –бітними шаблонами.	Можливість стиснення послідовності.
10	Ентропійний тест	Міра узгодженості значення ентропії джерела із тим, що теоретично очікується для випадкового джерела.	Нерівномірність розподілення m- бітних слів у послідовності (регулярність властивостей джерела)
11	Перевірка випадкових відхилень	Міра узгодженості кількості візитів при випадковому блуканні в заданий стан в середині циклу із тим, що очікується теоретично	Відхилення від теоретичного закону розподілення візитів у конкретний стан при випадковому блуканні
12	Перевірка випадкових відхилень (варіант)	Загальна кількість візитів при випадковому блуканні	Відхилення від теоретично очікуємої загальної кількості візитів при випадковому блуканні у заданий стан.
13	Послідовний тест	Міра узгодженості кількості усіх варіантів m-бітних шаблонів, що зустрілись, із тією, що очікується теоретично.	Нерівномірність розподілення m- бітних слів у послідовності.
14	Перевірка стиснення згідно алгоритму Лемпеля-Зіва	Кількість різних слів у послідовності	Великий ступінь стиснення послідовності, що тестується порівняно із ступенем стиснення, що очікується для випадкової послідовності.

15	Перевірка шаблонів, що не перекриваються	Міра узгодженості кількості неперіодичних шаблонів у послідовності із теоретичним значенням.	Велика кількість заданих неперіодичних шаблонів у послідовності.
16	Перевірка лінійної складності	Міра узгодженості кількості подій, що полягають у появі фіксованої довжини еквівалентного ЛРР для заданого блоку із теоретичним.	Відхилення емпіричного розподілу довжин еквівалентних ЛРР для послідовностей фіксованої довжини від теоретичного закону розподілення для випадкової послідовності, що вказує на недостатню складність послідовності, що тестується.

Таким чином у результаті тестування двійкової послідовності формується вектор значень імовірності $P=\left(P_1,P_2\ldots P_{189}\right)$. Аналіз складових P_i даного вектору дозволяє вказати на конкретні дефекти випадковості послідовності, що тестується.

Нижче коротко описано основні правила використання цього пакету для тестування статистичних характеристик різних послідовностей.
Для запуску графічної оболонки тестів використовують файл NIST_UI.exe.
Після його запуску на ерані виникне головна форма пакету, яка дозволяє керувати усіма його можливостями (рис.1).

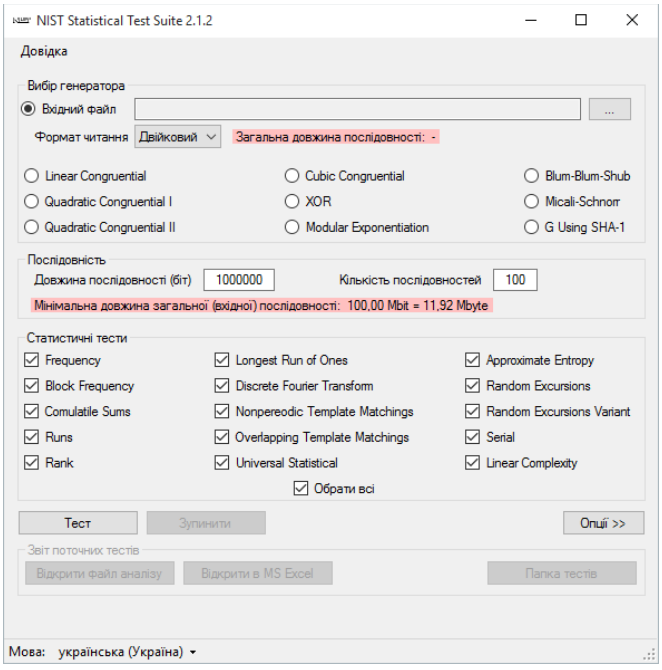


Рисунок 1: Головне вікно програми

Тут ми можемо вибрати генератор: або вхідний файл, або один з вбудованих генераторів для порівняння. Нас цікавить вхідний файл, тому ми натискаємо кнопку вибору файлів:

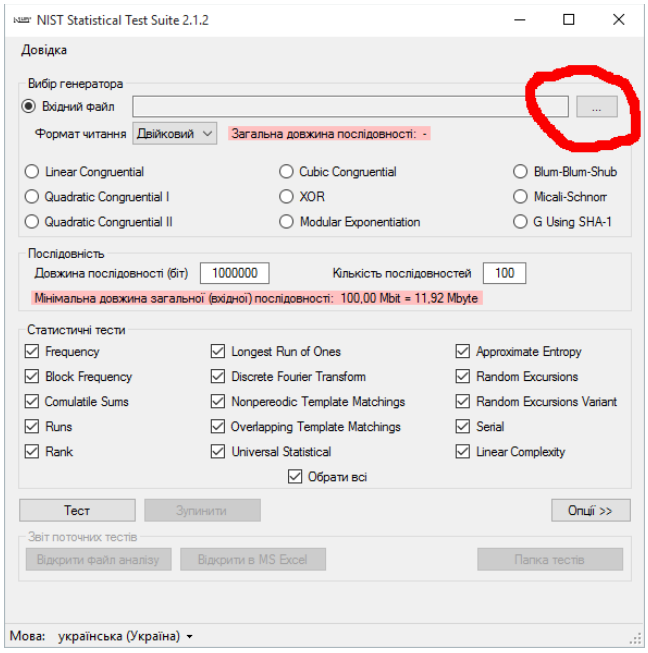


Рисунок 2: Кнопка вибору файлів

Якщо ми завантажили файл, згенерований шифром або генереатором послідовностей, який ми хочемо дослідити, система автоматично визначить, який він (двійковий або текстовий), яка його довжина та на скільки послідовностей його можна розбити. В результаті ми побачимо картинку, зображену на рис.3:

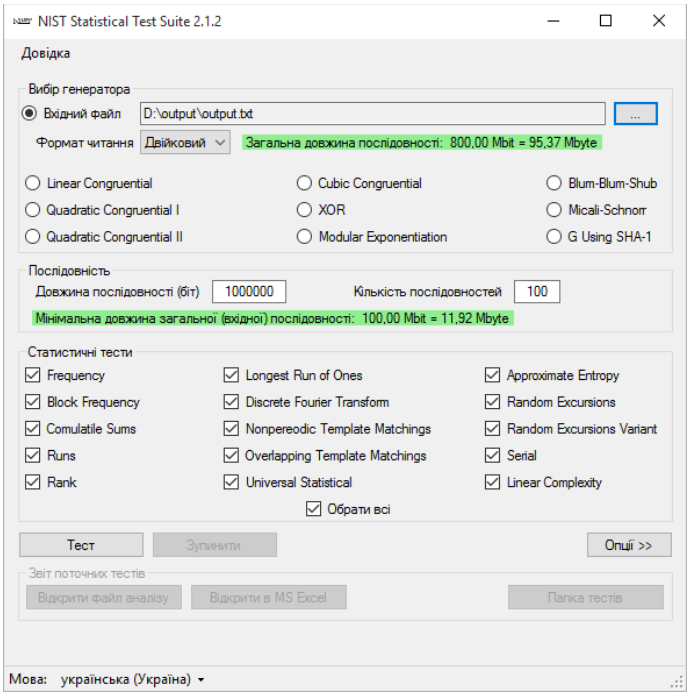


Рисунок 3: Система готова до тестування

Якщо повідомлення програми зафарбовано зеленим кольором, як на рисунку 3, система може протестувати вказаний файл. Якщо ж вони зафарбовані червоним — файл непридатний для тестування. В цьому випадку треба або вибрати інший файл, або перегенерувати його.

Для початку тестування натискаємо кнопку “Тест”. На формі з’явиться ProgressBar, який показує прогрес процесу тестування. По закінчення тестування система видасть повідомлення, як на рис.4.

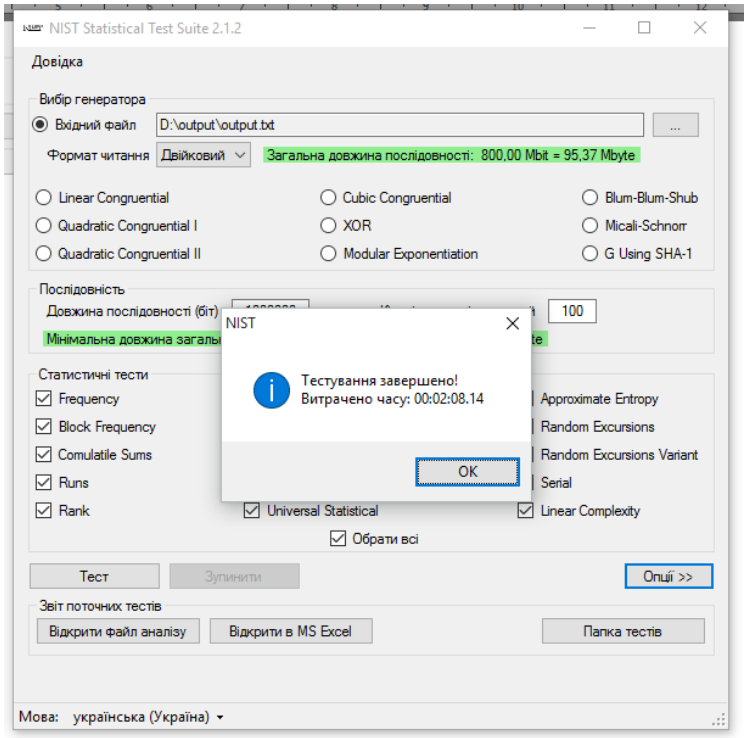


Рисунок 4: Повідомлення про завершення тестування

Тепер ми можемо подивитися на результати тестування. Це можливо зробити у два способи. Перший — експортувати результати в MS Excel. Тоді результати виглядатимуть як на рис.5.

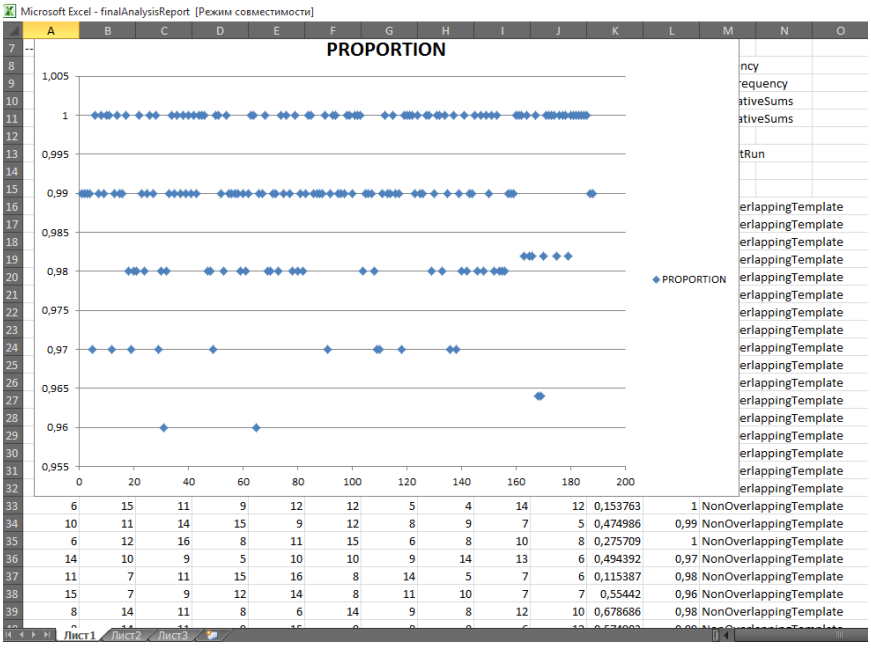
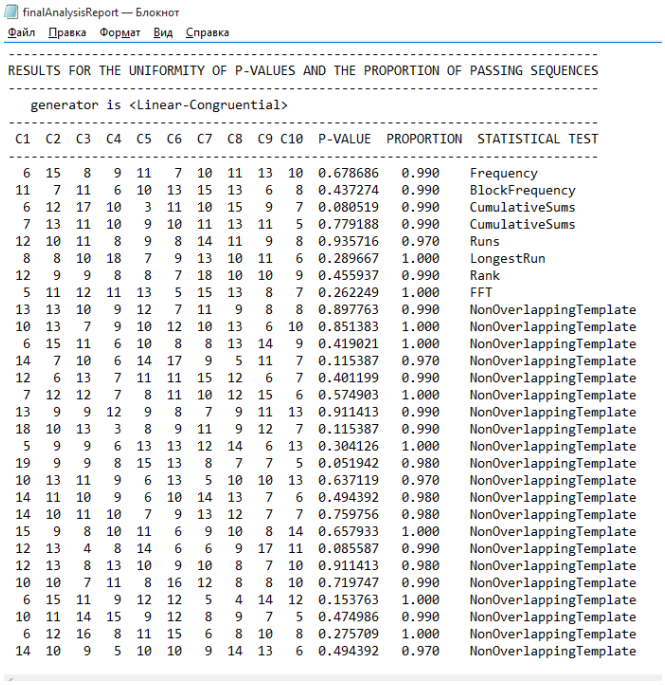


Рисунок 5: Результати тестування у MS Excel

Діаграму можна вставити безпосередньо у текст звіту з лабораторної роботи або зберегти на диску. Якщо ж ми не хочемо використати вбудований експорт, можна отримати текстовий файл результатів тестування, вигляд якого подано на рис.6.



finalAnalysisReport — Блокнот

Файл Правка Формат Вид Справка

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
   generator is <Linear-Congruential>
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  6  15   8   9  11   7  10  11  13  10  0.678686   0.990    Frequency
 11   7  11   6  10  13  15  13   6   8  0.437274   0.990    BlockFrequency
  6  12  17  10   3  11  10  15   9   7  0.080519   0.990    CumulativeSums
  7  13  11  10   9  10  11  13  11   5  0.779188   0.990    CumulativeSums
 12  10  11   8   9   8  14  11   9   8  0.935716   0.970    Runs
  8   8  10  18   7   9  13  10  11   6  0.289667   1.000    LongestRun
 12   9   9   8   8   7  18  10  10   9  0.455937   0.990    Rank
  5  11  12  11  13   5  15  13   8   7  0.262249   1.000    FFT
 13  13  10   9  12   7  11   9   8   8  0.897763   0.990    NonOverlappingTemplate
 10  13   7   9  10  12  10  13   6  10  0.851383   1.000    NonOverlappingTemplate
  6  15  11   6  10   8   8  13  14   9  0.419021   1.000    NonOverlappingTemplate
 14   7  10   6  14  17   9   5  11   7  0.115387   0.970    NonOverlappingTemplate
 12   6  13   7  11  11  15  12   6   7  0.401199   0.990    NonOverlappingTemplate
  7  12  12   7   8  11  10  12  15   6  0.574903   1.000    NonOverlappingTemplate
 13   9   9  12   9   8   7   9  11  13  0.911413   0.990    NonOverlappingTemplate
 18  10  13   3   8   9  11   9  12   7  0.115387   0.990    NonOverlappingTemplate
  5   9   9   6  13  13  12  14   6  13  0.304126   1.000    NonOverlappingTemplate
 19   9   9   8  15  13   8   7   7   5  0.051942   0.980    NonOverlappingTemplate
 10  13  11   9   6  13   5  10  10  13  0.637119   0.970    NonOverlappingTemplate
 14  11  10   9   6  10  14  13   7   6  0.494392   0.980    NonOverlappingTemplate
 14  10  11  10   7   9  13  12   7   7  0.759756   0.980    NonOverlappingTemplate
 15   9   8  10  11   6   9  10   8  14  0.657933   1.000    NonOverlappingTemplate
 12  13   4   8  14   6   6   9  17  11  0.085587   0.990    NonOverlappingTemplate
 12  13   8  13  10   9  10   8   7  10  0.911413   0.980    NonOverlappingTemplate
 10  10   7  11   8  16  12   8   8  10  0.719747   0.990    NonOverlappingTemplate
  6  15  11   9  12  12   5   4  14  12  0.153763   1.000    NonOverlappingTemplate
 10  11  14  15   9  12   8   9   7   5  0.474986   0.990    NonOverlappingTemplate
  6  12  16   8  11  15   6   8  10   8  0.275709   1.000    NonOverlappingTemplate
 14  10   9   5  10  10   9  14  13   6  0.494392   0.970    NonOverlappingTemplate
```

Рисунок 6: Результати тестування у текстовому вигляді

Як оцінити отримані результати? NIST STS розділяє вхідний файл (якщо його довжина дозволяє) на 100 однакових підпослідовностей по 1 млн бітів кожна. Тому довжина вхідного файлу повинна буди 100Мб (12,5 МБ). До кожної з цих підпослідовностей застосовуються усі 189 статистичних тестів. Вважається, що повна послідовність пройшла конкретний тест, якщо хоча би 96 зі 100 підпослідовностей його пройшли. Тому найкращим вважається той шифр/генератор, вихідна послідовність якого пройшла усі тести з максимальною пропорцією.

Часто результати тестування показують як табличку, де вказано, скільки тестів NIST пройдено на якому рівні (див. Табл.2):

Таблиця 2. Результати порівняльного тестування 5 шифрів у табличному вигляді.

Ймовірність проходження тестів, %	Шифр1	Шифр2	Шифр3	Шифр4	Шифр5
	Кількість тестів, які пройшли тестування (%)				
100	81 (43%)	89 (47,1%)	129 (68,3%)	80 (42,5%)	70 (37%)
99	58 (31%)	65 (34,4%)	16 (8,5%)	57 (30,5%)	60 (32%)
98	36 (19%)	26 (14%)	23 (12,2%)	34 (18%)	43 (23%)
97	12 (6,5%)	6 (3,2%)	8 (4,2%)	13 (7%)	13 (7%)
96	1 (0,5%)	2 (1%)	2 (1%)	2 (1%)	2 (1%)
95	-	-	3 (1,6%)	2 (1%)	-
< 95	-	-	6 (3,2%)	-	-
Середнє значення	0,98571 4286	0,98714 2857	0,981904 762	0,98507 9365	0,984497 354

Ще одним простим методом порівняльної оцінки шифрів є середнє значення ймовірності проходження тестів. Дуже наближено можна вважати, що шифр/генератор, який має найбільше середнє значення ймовірності проходження тестів, має найкращі статистичні характеристики. З такої точки зору (в середньому) найкращим можна вважати Шифр 1, оскільки його середнє значення найбільше.

Практична частина

1. Студент обирає два шифри з наявних у криптобібліотеці або один шифр та один генератор псевдовипадкових двійкових послідовностей.
2. Студент розробляє систему шифрування на основі обраних примітивів, яка має задовольняти такі вимоги:
- шифр має приймати на вхід файл розіром 12,5 МБ (100 Мб) і шифрувати його у текстовий або бінарний файл (текстовий повинен складатися з нулів та одиниць);
- генератор повинен генерувати псевдовипадкову послідовність у файл, розміром 100 Мб (тобто він повинен згенерувати 100 млн бітів).
3. Отримані файли подаються на вхід NIST STS та повинні пройти статистичне тестування.
4. Результати подаються за бажанням студента у вигляді діаграми MS Excel або табличному (як показано в Табл.2).
5. За результатами досліджень робиться висновок про криптографічну стійкість розроблених систем шифрування.
6. Сформуйте звіт з лабораторної роботи, який має містити:
- протокол Ваших дій;
- відкритий та зашифрований текст;
- результати тестування шифрів/генераторів;
- висновки про їх криптостійкість;
- код програми;
- відповіді на контрольні запитання.

Код програми має задовольняти вимоги до написання коду (наприклад, C++ Programming Style Guideline).

Контрольні запитання

1. Призначення та основні можливості NIST STS.
2. Які характеристики визначають тести NIST STS? Для чого це потрібно?
3. Яка методика тестування у NIST STS?
4. Як можна порівняти криптостійкість шифрів за допомогою NIST STS?